Projektarbeit Sommer 2001



Erstellung eines Lösungspakets zur Benutzerverwaltung auf einem Linux-Server

Name:	Tobias Brinkert
Ausbildungsberuf:	Fachinformatiker FR Systemintegration
Ausbildungsbetrieb:	Bildungszentrum Dortmund
Abgabedatum:	30.03.2001
Azub.-Identnr.:	xxxxxx

Inhaltsverzeichnis

1 Beschreibung des Auftrages

1.1 Problemstellung

Der Auftraggeber, die Realschule Ascheberg, betreibt zwei EDV-Räume im zweiten Hauptgebäude. Als primärer File-Server dient eine Linux-Maschine mit einer SuSE-Distribution Version 7.0 und Samba Version 2.0.7.

Für den Zugang zum Netzwerk soll jeder Schüler und Lehrer einen eigenen Account erhalten und Speicherplatz auf dem primären File-Server zugewiesen bekommen. Mit der Entwicklung eines Lösungspaketes zur Benutzerverwaltung wurde ich von Herrn Schubert, Schulleiter der Realschule Ascheberg, betraut. Dabei sollen die folgenden Funktionen abgedeckt werden:

- Benutzer- sowie Samba-Accounts anlegen, modifizieren und löschen;
- Gruppen anlegen, modifizieren und löschen;
- eine automatische Benutzeranlegung anhand einer Schüler- bzw. Lehrerliste mit Speicherung der Benutzerdaten;
- Paßwörter sollen für Schüler automatisch generiert werden;
- Möglichkeit der Fernadministration mittels Webbrowser.

Das zu entwickelnde Produkt soll auf dem primären File-Server der Realschule Ascheberg eingesetzt werden. Zusätzlich soll die Möglichkeit bestehen, daß das Lösungspaket in anderen Schulen mit Linux-Servern eingesetzt werden kann. Hierzu ist die Implementierung von Mechanismen zur einfachen Anpassung an die Systemgegebenheiten und -richtlinien der einzelnen Schulen notwendig.

1.2 Projektumfeld

In der Realschule Ascheberg wird derzeit die informationstechnische Infrastruktur aufgerüstet, die im Rahmen eines Projektes unserer Umschulungsklasse abläuft. Neben den beiden Informatikräumen, die sich im zweiten Hauptgebäude befinden, sollen alle Klassen- und Fachräume einen Zugang zum Netzwerk erhalten. Hierfür sind zwei Linux File-Server und ein Linux Proxy-Server - eine von Herrn Lothar Dornieden entwickelte Komplettlösung für Schulen - vorgesehen, die Ressourcen zur gemeinsamen Nutzung zur Verfügung stellen. Als Clients kommen überwiegend Windows-Maschinen zum Einsatz. Die Koppelung der beiden Hauptgebäude übernehmen die beiden Linux File-Server, die physikalische Konnektierung erfolgt über eine Lichtwellenleiterverbindung. Geplant ist die Fertigstellung der Vernetzungsarbeiten für Ende März.

1.3 Technische Schnittstelle und Implementierung

Anwender und Lösungspaket sollen über einen Webbrowser miteinander kommunizieren. Auf dem Server muß hierfür entweder ein HTTP-Server (z.B. Apache) installiert sein oder das Lösungspaket als Modul in die Administrationsoberfläche WebMin integriert werden. Beide Wege werden im Projektverlauf untersucht und gegeneinander abgewogen, um eine Entscheidung für einen Lösungsweg zu treffen.

Der Zeitpunkt der Einbindung des Produktes ins Netzwerk wird mit dem Schulleiter abgesprochen. Bei der Entwicklung wird darauf Wert gelegt, daß zur Implementierung auf dem primären File-Server so wenig Modifikationen wie möglich durchgeführt werden müssen. Eine Absprache mit dem System-Administrator ist nicht notwendig, da der Entwickler selbst den Server aufgesetzt hat und für die Administration verantwortlich ist.

1.4 Änderungen gegenüber dem Projektantrag

Im zweiten Kundengespräch mit Herrn Schubert wurde deutlich, daß die klassenweise und dauerhafte Speicherung der generierten Benutzerdaten (Accountname, Paßwort ...) auf einem nur für Lehrer zugänglichen Serverbereich - durch das Überwinden der Sicherheitsmechanismen oder Fahrlässigkeit der Lehrer - Probleme mit der Zuordnung von Netzbenutzung und Benutzer verursachen kann. Daraufhin werden die Benutzerdaten nur noch in einer Datei erfolgen. Auf Basis dieser Datei wird ein Serienbrief mit den Benutzerdaten erstellt und die Datei anschließend gelöscht (muß vom Anwender durchgeführt werden).

2 Konzepterstellung und Planung

2.1 Ist-Analyse

Zur Zeit werden für den Netzwerk-Zugriff globale Benutzer und Kennwörter verwendet. Ausschließlich die Informatik-Lehrer, der Schulleiter und der System-Administrator besitzen eigene Accounts auf dem primären File-Server. Der Server wird mit SuSE-Linux 7.0 betrieben, wobei unter anderem die Dienste Samba und HTTP-Server (Apache) bereitgestellt werden. Zur Administration wird das Tool WebMin über eine verschlüsselte Verbindung (SSL) eingesetzt. Für den Einsatz von WebMin ist eine Perl-Distribution erforderlich, die in der Version 5.005_03 installiert ist.

Alle Clients der Schule verfügen über einen JavaScript-fähigen Webbrowser von Netscape. Eine FileMaker Pro-Datenbank mit den Schülerdaten befindet sich in der Verwaltung. Hieraus können die Felder Vorname, Nachname und Klasse als relevante Daten im Macintosh-Format exportiert werden. Erwartet wird ein Datensatzumfang von ca. 550 Schülern.

2.2 Soll-Konzept

Mit dem zu entwickelnden Lösungspaket soll die Möglichkeit bestehen, die Benutzer und Gruppen effektiv zu verwalten, Gruppenverzeichnisse anzulegen sowie zum Schuljahresanfang alle Schüler schnell und einfach automatisch anzulegen und ein Paßwort zuzuweisen. System- und Samba-Accounts sollen dabei in einem Schritt angelegt, modifiziert oder gelöscht werden können. Die Administrationsoberfläche soll einfach zu erlernen und selbstbeschreibend sein. Spezielle Linux-Kenntnisse sollen für die Benutzer- und Gruppenverwaltung normalerweise nicht erforderlich sein. Zusätzlich muß das Lösungspaket konfigurierbar sein, die Implementierung ohne größere Systemmodifikationen und Ausfallzeiten auf einem Linux-System erfolgen und keine zusätzlichen Kosten für kommerzielle Software erfordern, damit die Lösung auch in anderen Schulen eingesetzt werden kann.

Das Lösungspaket soll zum 15. März 2001 fertiggestellt und auf dem File-Server der Realschule Ascheberg implementiert werden.

2.3 Systemintegrationskonzept

Für den Systementwurf wurden die beiden Lösungswege, die Implementierung über den HTTP-Server oder als Modul in das Administrationsprogramm WebMin, gegenübergestellt. Die Gegenüberstellung ist in Tabelle 1 wiedergegeben.

HTTP-Server		**WebMin**	
pro	***kontra***	***pro***	***kontra***
bei jeder Linux-Version mitgeliefert	Authentifizierung mit Accounts muß eingerichtet werden	bekannte und einheitliche Administrationsumgebung	muß meist separat installiert werden
unterstützt Script sowie Compilersprachen	Verschlüsselte Verbindung muß gesondert eingerichtet werden	Administrator-Operationen sind erlaubt	keine Compilersprachen wegen Portierbarkeit
	standardmäßige Unterbindung von Administrator-Operationen	Verschlüsselte Verbindung ist einfach zu realisieren	Design vorgegeben
		unterstützt Scriptsprachen	
		Mechanismen zur Konfigurierbarkeit, Hilfe und Internationalisierung vorhanden	
		leichte Installation und Update-Möglichkeit	

Tabelle 1: Gegenüberstellung HTTP-Server vs. WebMin

Aufgrund des Pro-Kontra-Vergleichs wurde eine Implementierung als Modul in das Administrationsprogramm WebMin angestrebt. Dieses begründet sich vor allem im geringeren Aufwand für Systemmodifikationen und der Sicherheit (SSL-Verbindung sowie die Erhaltung des Wrappers). Zusätzlich wird für das Produkt eine höhere Akzeptanz und schnellere Erlernbarkeit erwartet, da WebMin schon auf allen Servern eingesetzt wird und eine einheitliche Administrationsoberfläche geschaffen wird. Um die WebMin API verwenden zu können, wird für die Implementierung die Interpreter-Sprache Perl eingesetzt.

2.4 Pflichtenheft

Das Pflichtenheft ist vollständig als Anlage in Kapitel 7, Anlage 1, einzusehen.

2.5 Programmentwurf

Der notwendige Programmentwurf wurde als Sitemap entwickelt und ist im Anhang, Kapitel 6.1, beigefügt. Diese diente als Grundlage für die programmiertechnische Realisierung.

3 Entwicklung

3.1 Voraussetzungen

Voraussetzung für die Entwicklung ist ein Linux-Server und ein Client mit Webbrowser. Um die spätere Produktumgebung nachbilden zu können, wurde auf den Server eine SuSE Linux-Distribution Version 7.0, WebMin Version 0.83 und Perl Version 5.005_03 aufgespielt. Der Client wurde mit SuSE Linux Version 7.0, WebMin Version 0.81 und Perl Version 5.005_03 betrieben, da in der Testphase das Produkt auf dieser Maschine zusätzlich installiert und getestet werden sollte. Als Webbrowser kam Netscape Navigator Version 4.75 zum Einsatz.

3.2 Entwicklungsphasen

Die Entwicklung des Lösungspaketes wurde in 4 Phasen aufgeteilt, die sich anhand der Funktionalität bildeten. Zuerst wurde die Startseite nur mit der grundlegenden Funktionalität ausgestattet, damit die einzelnen Modul-Komponenten gestartet werden können. Während der gesamten Entwicklungszeit wurde eine Funktionsbibliothek aufgebaut, die von mehreren Scripten benötigt werden, und der Konfigurationsbereich implementiert. Konfigurationsbereich, Textausgabe, Hilfe und Design wurden an die Vorgaben und Methoden von WebMin angepaßt. Um die Überprüfungen der Konfigurationseinstellungen zu verringern, wurde nur auf der Startseite die Möglichkeit der globalen Konfiguration gegeben. Jedes Script übernimmt die Überprüfung der übergebenen Werte sowie die Fehlerbehandlung. Desweiteren gliederte sich die Entwicklung in folgende 4 Phasen.

3.2.1 Gruppenmanager

Begonnen wurde mit der Entwicklung des Gruppenmanagers, da die Benutzerverwaltung logisch hiervon abhängig ist. Der Gruppenmanager (siehe Abbildung 1b) gliedert sich in die Teile

1. Eingabefeld und Optionen, die für die Erstellung einer neuen Gruppe notwendig sind,
2. eine Tabelle mit vorhandenen (normalen) Gruppen und
3. eine Tabelle mit vorhandenen System-Gruppen.

Über den ersten Teil kann eine neue Gruppe spezifiziert und durch Betätigen des Button "Gruppe anlegen" die Gruppe angelegt werden. Hier wurde die Möglichkeit implementiert, ein Gruppenverzeichnis anzulegen. Der Ort des Gruppenverzeichnisses kann über ein Dropdown-Feld angegeben werden, wobei die Pfade "Schüler-Heimatverzeichnis + Gruppenname + Gruppenname (Standard)", "Lehrer-Heimatverzeichnis + Gruppenname" und "Gruppen-Verzeichnis + Gruppenname" zur Verfügung stehen. Die Zugriffsrechte für das neue Gruppenverzeichnis werden über die Konfigurationseinstellungen des Moduls eingestellt.

Beide nachfolgenden Tabellen mit den vorhandenen Benutzer- und Systemgruppen bieten die Möglichkeit, Gruppen zu modifizieren bzw. Detailinformationen anzeigen zu lassen und Gruppen aus dem System zu entfernen. Unterschieden werden dabei die beiden Gruppenarten anhand ihrer Gruppen-Identifikationsnummer (GID), die bei Systemgruppen einen Wert unter 100 oder den Wert 65534 besitzen.

Detailinformationen und Modifizierungen zu einer Gruppe wurden über einen Link auf den Gruppennamen realisiert. Die Modifizierung beschränkt sich auf die Verwaltung von sekundären Gruppenmitgliedern.

Eine oder mehrere Gruppen können über den Button "Gruppen löschen" aus dem System entfernt werden. Die Auswahl der zu löschenden Gruppen wird über die Checkboxen rechts neben dem Gruppennamen in den beiden Gruppentabellen vorgenommen. Zu löschende Gruppen dürfen keine primären Gruppenmitglieder enthalten, ansonsten wird diese Gruppe nicht gelöscht. Ein gleichzeitiges Löschen des Gruppenverzeichnisses ist nicht implementiert, da die Systemdatenbanken eine solche Information nicht speichern. Der Bedarf an einer solchen Funktionalität wird als sehr gering eingestuft, so daß auf eine Implementierung einer eigenen Datenbank verzichtet wurde.

3.2.2 Benutzermanager

Für den Benutzermanager (siehe Abbildung 1a) wurde derselbe Aufbau wie beim Gruppenmanager gewählt, um eine einheitliche Oberfläche sowie eine einfache Einarbeitung



(a) Benutzermanager



(b) Gruppenmanager

Abbildung 1: Benutzer- und Gruppenmanager

zu bieten. Die Optionen für einen neuen Benutzer sind aufgrund der Systemgegebenheiten weitreichender als die einer neuen Gruppe. Die wichtigsten Funktionen, die die vorhandene Benutzerverwaltung von WebMin nicht bietet, ist das gleichzeitige Anlegen eines Samba-Accounts sowie die Möglichkeit der Generierung eines Paßwortes (siehe auch Kapitel 3.3). Anhand der Einstellung, ob es sich bei dem anzulegenden Benutzer um einen Schüler oder Lehrer handelt, wird der Pfad des Heimatverzeichnisses gebildet. Dieser setzt sich bei Schülern aus dem Pfad zu den Schüler-Heimatverzeichnissen, der primären Gruppe und dem Anmeldenamen zusammen, bei Lehrern aus dem Pfad zu den Lehrer-Heimatverzeichnissen und dem Anmeldenamen.

Bei der Benutzermodifizierung besteht neben der Modifizierung der Gruppenmitgliedschaften die Möglichkeit, das Paßwort sowohl für den System-Account als auch für den Samba-Account zu verändern. Zusätzlich kann der Samba-Account modifiziert und die zugeordnete Shell verändert werden.

Für das Löschen von Benutzern gibt es bei normalen Benutzern die Option, das Heimatverzeichnis zu entfernen. Bei System-Benutzern wurde auf diese Option verzichtet, da sich das Heimatverzeichnis von solchen Benutzern häufig im normalen Verzeichnisbaum befindet. Hierdurch sollen Funktionseinschränkungen oder Systemausfälle durch Löschen von wichtigen Systemverzeichnissen vermieden werden. Die Zuordnung zu normalen Benutzern bzw. System-Benutzern wird anhand der Benutzer-Identifikationsnummer (UID) vorgenommen. Benutzer mit einer UID kleiner 100 oder einer UID von 65534 werden den System-Benutzern zugeordnet, alle anderen den normalen Benutzern.

3.2.3 Automatische Benutzeranlegung

Die automatische Benutzeranlegung gliedert sich in zwei Abschnitte, die Konfigurationseinstellung (siehe Abbildung 2a) und das eigentliche Anlegen. Dabei werden in den Konfigurationseinstellungen die benötigten Werte, wie z.B. Datei mit Benutzerdaten, Pfad für die Datei mit den generierten Benutzerdaten, Typ des Benutzers (Schüler oder Lehrer) angegeben. Anschließend kann über den Button "Benutzer anlegen" die automatische Benutzeranlegung gestartet werden.

Erforderliche Felder in der Benutzerdaten-Datei sind Vorname, Nachname und Gruppe, wobei sich in jeder Zeile der Datei nur ein Datensatz befinden darf. Mögliche Dateiformate sowohl für die Import- als auch für die Export-Datei sind Linux, Macintosh, DOS und MS Windows. Umgesetzt ins Linux-Format werden diese mit Hilfe des Programms "recode". Als Feldtrennzeichen können Kommata, Semikola, Doppelpunkte oder Rauten (Nummernzeichen) eingesetzt werden, wobei nur eines dieser Feldtrennzeichen in einer Import-Datei verwendet werden darf. In der Export-Datei wird das gleiche Feldtrennzeichen wie in der Import-Datei verwendet.

Das Script zur automatischen Benutzeranlegung überprüft zuerst die übergebenen Werte und das Vorhandensein der benötigten Gruppen. Für die Gruppennamen kann ein Prefix in den Konfigurationseinstellungen angegeben werden. Sollte hier ein Wert eingetragen sein und der Gruppenname in der Import-Datei nur aus einer Ziffer und einem Buchstaben bestehen, wird der Ziffer eine "0" vorangestellt, um eine zweistellige Ziffer zu erhalten.

Der Accountname wird aus Vor- und Nachname gebildet, wobei dieser maximal 12 Zeichen lang sein darf und in Kleinbuchstaben umgewandelt wird. Dabei wird zuerst versucht, einen Accountnamen aus dem 1. Buchstaben des Vornamens und dem Nachnamen zu bilden, danach aus den ersten beiden Buchstaben des Vornamens und dem Nachnamen, usw (siehe Struktogramm im Anhang, Kapitel 6.2.2). Sollte kein eindeutiger Anmeldename generiert werden können, wird der Benutzer nicht angelegt.

Bei der automatischen Benutzeranlegung wird für jeden Benutzer ein Heimatverzeichnis mit dem gleichen Algorithmus wie bei der manuellen Benutzeranlegung (siehe Kapitel 3.2.2) angelegt, ein Paßwort automatisch generiert, ein Samba-Account eingerichtet sowie eine direkte Anmeldung auf dem System verhindert. Die generierten Benutzerdaten werden in die Export-Datei geschrieben.

3.2.4 Startseite

Als letzter Entwicklungsteil wurde die Startseite implementiert. Die Startseite (siehe Abbildung 2b) übernimmt - zusätzlich zur Startmöglichkeit der drei Programmteile - die Überprüfung der Konfigurationseinstellungen sowie die Überprüfung, ob alle benötigten Systemprogramme verfügbar sind. Bei Entwicklungsbeginn waren noch nicht alle Konfigurationswerte und externe Systemprogramme bekannt, aus diesem Grund wurde dieser

Entwicklungsschritt ans Ende verschoben. Die Konfiguration ist nur über diese Startseite erreichbar, so daß die Überprüfung der Konfigurationsparameter nur in diesem Bereich notwendig ist.



(a) Automatische Benutzeranlegung



(b) Startseite

Abbildung 2: Automatische Benutzeranlegung und Startseite

3.3 Automatische Paßwortgenerierung

Gewünscht war die Möglichkeit der automatischen Paßwortgenerierung. Dabei sollte das Paßwort nicht aus einem Wörterbuch, sondern durch eine zufällige Zeichenfolge generiert werden. Hierfür wurde ein Algorithmus entwickelt, welcher ein Paßwort aus 8 Zeichen generiert. Für die ersten 4 Zeichen wird versucht, den Anmeldenamen mit Hilfe des Soundex-Algorithmus (gehört zur Perl-Distribution) zu packen und mit 4 zufälligen Zeichen aufzufüllen. Sollte der Anmeldename nicht gepackt werden können, wird das gesamte Paßwort aus zufälligen Zeichen generiert. Das Struktogramm hierfür ist im Anhang, Kapitel 6.2.1, einzusehen.

3.4 Sicherheits- und Zugriffskonzept

Ein eigenständiges Sicherheits- und Zugriffskonzept ist für das entwickelte Lösungspaket nicht notwendig, da über WebMin diese Methoden schon implementiert sind. Dabei ist, durch die Übertragung von Paßwörtern, die Verwendung einer verschlüsselten Verbindungen zwischen Server und Client anzuraten, die auf dem primären File-Server mittels einer SSL-verschlüsselten WebMin-Sitzung schon implementiert ist.

Desweiteren werden die Veränderungen der Systemdatenbanken nicht direkt durchgeführt, sondern über die entsprechend dafür vorgesehenden Programme, so daß keine Methoden zum Locking gesondert implementiert werden müssen. Hiermit soll sichergestellt

werden, daß keine Beschädigungen an den Systemdatenbanken durch die parallele Verwendung dieses WebMin-Moduls eintreten können.

3.5 Qualitätssicherung und Freigabe

Um ein fehlerfreies Produkt zu garantieren, wurden alle Funktionen des WebMin-Moduls überprüft. Als Clients kamen folgende Konstellationen zum Einsatz:

1. SuSE-Linux Version 7.0 mit dem Webbrowser Netscape Navigator 4.75 und den Auflösungen 800 x 600 dpi, 1024 x 786 dpi und 1280 x 1024 dpi;
2. MS Windows 98 mit den Webbrowsern Netscape Communicator 4.75, MS Internet-Explorer 4.0 und 5.5 sowie den Auflösung: 800 x 600 dpi und 1024 x 786 dpi;
3. MacOS 7.6.1 mit dem Webbrowser Netscape Navigator 3.01 und einer Auflösung von 640 x 480 dpi.

Daneben wurde das Lösungspaket auf dem Entwicklungs-Client installiert und alle Funktionen mit den oben genannten Client-Konstellationen noch einmal überprüft. Beim Macintoshsystem zeigten sich aufgrund der geringen Bildschirmauflösung Schwächen in der Übersichtlichkeit. Die Einsatzfähigkeit des Produkts ist hiervon aber nicht betroffen.

Freigegeben wurde das Produkt mit der Versionskennung 0.2.0. Die Implementierung auf dem File-Server erfolgte am 1. März 2001 nach Absprache mit dem Schulleiter Herrn Schubert. Modifizierungen am System waren nicht notwendig. Auf dem primären File-Server wurden die notwendigen Modulkonfigurationen durchgeführt und die Funktionalität mit den vorhandenen Clients (Windows 95, Netscape Communicator 4.73) überprüft.

4 Soll-Ist-Vergleich

In der Tabelle 2 ist ein Rückblick zwischen den Muß- und Wunschkriterien, die im Pflichtenheft definiert worden sind, und der umgesetzten Funktionalität des Benutzer- und Gruppenmanagers wiedergegeben. Ein Soll-Ist-Vergleich der Zeitplanung ist über die Tabelle 3 durchgeführt worden. Dabei zeigt sich, daß die Umsetzung des Lösungspaketes als WebMin-Modul die Zeit für den Anwendungsentwurf sowie Installation und Testlauf verkürzten. Die Dauer für die Einweisung der EDV-Lehrer konnte durch die bekannte Oberfläche verringert werden, jedoch ist für die umfangreiche Dokumentation mehr Zeit benötigt worden. Durchgeführt wurde das Projekt innerhalb der veranschlagten 35 Stunden vom 12. Feb. bis 23. Feb. 2001. Die Vorstellung des Produktes sowie die Einweisung der EDV-Lehrer erfolgte am 05. März 2001.

Kriterium	Muß [M] / Wunsch [W]	Implementiert
Benutzerverwaltung	M	ja
Gruppenverwaltung	M	ja
automatische Account-Anlegung	M	ja
Samba-Accounts verwalten	M	ja
Apple Macintosh-Dateien verarbeiten	M	ja
Windows-Dateien verarbeiten	W	ja
DOS-Dateien verarbeiten	-	ja
Netzwerkfähigkeit (WebMin-Modul)	M	ja
Konfigurierbarkeit / Anpassungsfähigkeit	W	ja
Einsatz auf anderen Linux-Distributionen als SuSE Linux	W	nein

Tabelle 2: Soll-Ist-Vergleich der Zielbestimmung (siehe Kapitel 7, Anlage 1)

Projektphase	Soll-Zeit	Ist-Zeit
Kundengespräch	1,0 Std.	0,5 Std.
Problemanalyse	1,5 Std.	1,0 Std.
Anwendungsentwurf	4,0 Std.	3,0 Std.
Kundengespräch	0,5 Std.	0,5 Std.
Entwicklung der Oberfläche mit Benutzerverwaltung	9,0 Std.	10,0 Std.
Entwicklung des Scripts zur automatisierten Account-Anlegung	4,0 Std.	4,0 Std.
Funktions- und Leistungsprüfung	3,0 Std.	2,0 Std.
Installation und Testlauf	1,0 Std.	0,5 Std.
Einweisung der EDV-Lehrer	1,0 Std.	0,5 Std.
Projekt-, Benutzer- und System-Dokumentation	10,0 Std.	13,0 Std.
Gesamt	**35,0 Std.**	**35,0 Std.**

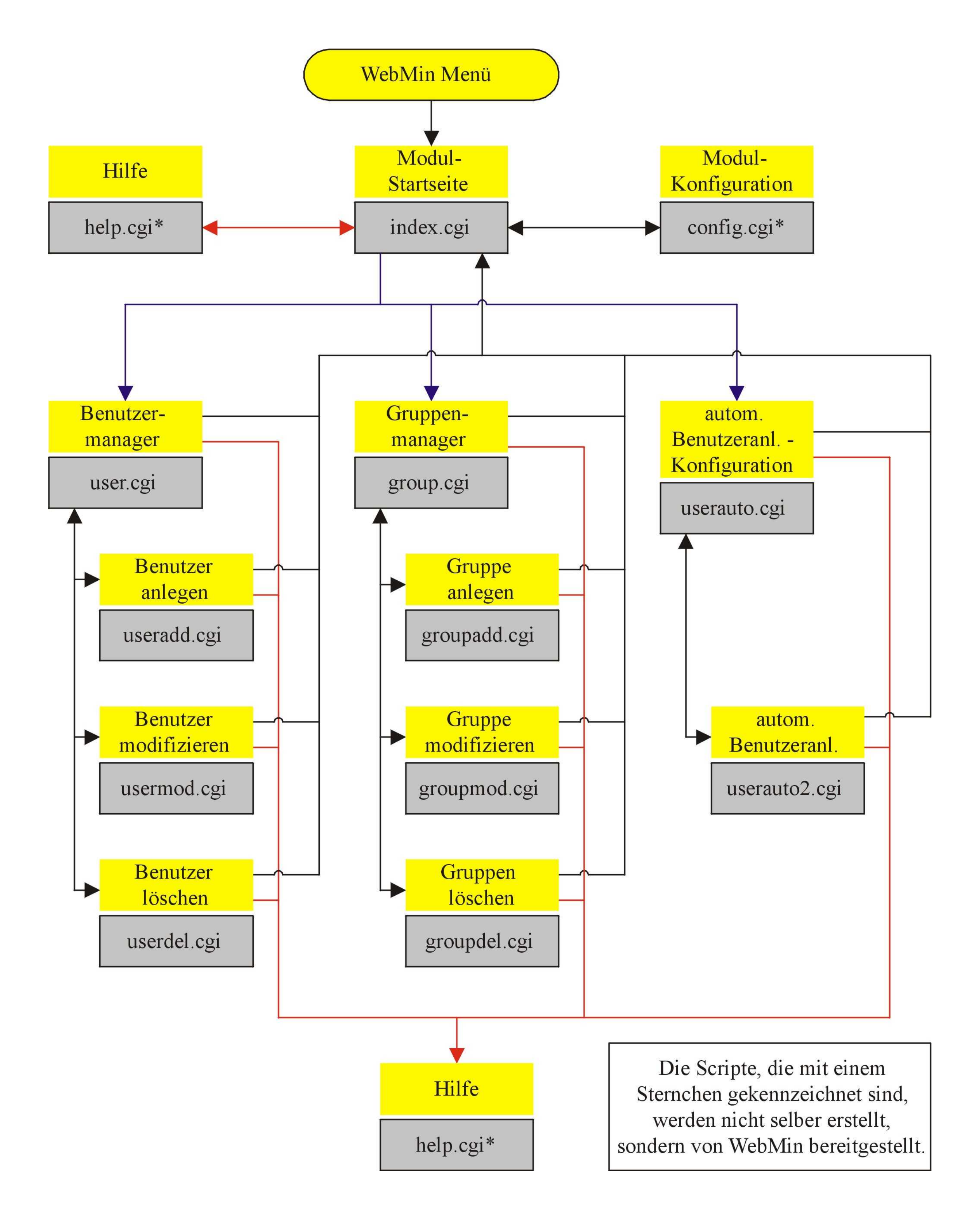
Tabelle 3: Soll-Ist-Vergleich der Zeitplanung

5 Ausblicke

Die Rechte an dem entwickelten WebMin-Modul bleiben vollständig beim Entwickler, so daß spätere Weiterentwicklungen möglich sind. Dabei wird das Modul unter die GPL-Lizenz und im Internet der Öffentlichkeit zur Verfügung gestellt. Zusätzlich wird das WebMin-Modul in den von Herrn Lothar Dornieden für die Schulen des Kreis Coesfeld entwickelten Kommunikationsserver Sirius implementiert.

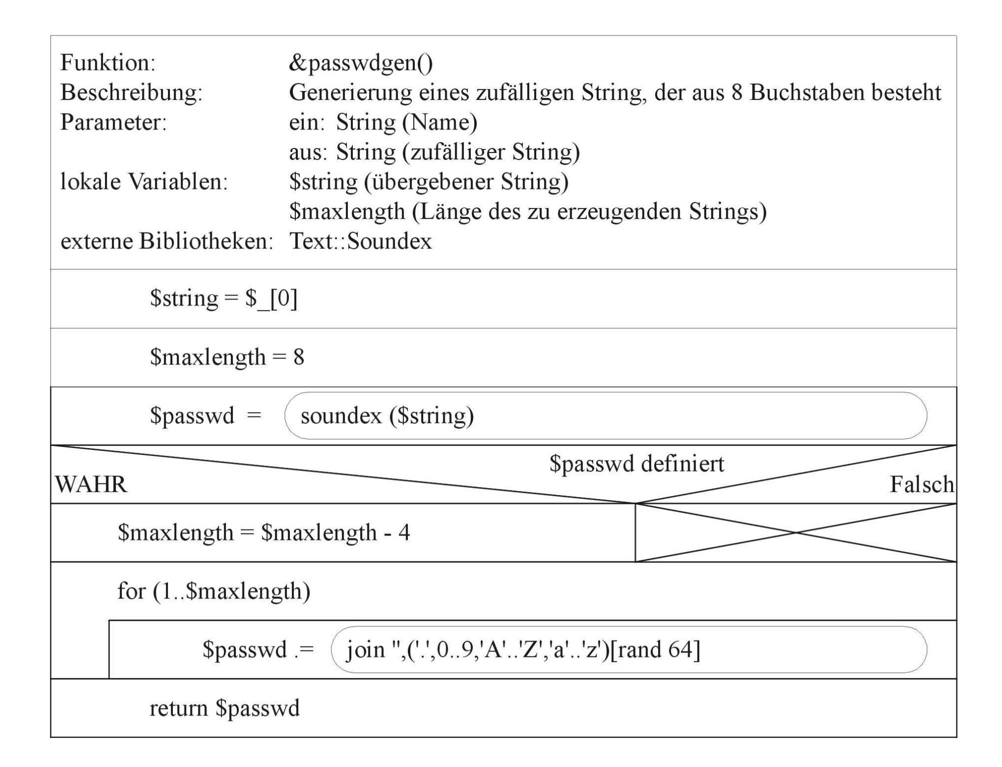
6 Anhang

6.1 Sitemap

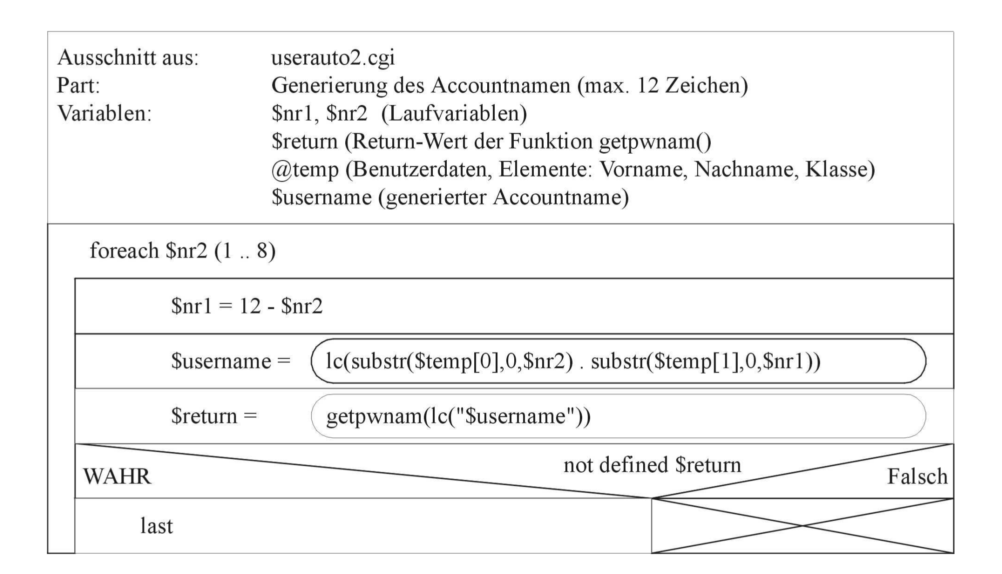


6.2 Struktogramme

6.2.1 Funktion zur Paßwortgenerierung aus der Bibliothek usermgr-lib.pl



6.2.2 Ausschnitt “Generierung des Anmeldenamens” aus userauto2.cgi



6.3 Glossar

primäres Gruppenmitglied: Benutzer, die eine Gruppe als Hauptgruppe verwenden und mir ihr bei der Anmeldung initialisiert werden sind primäres Gruppenmitglied dieser Gruppe. Alle Dateien, die der Benutzer neu anlegt, werden der primären Gruppe des Besitzers zugeordnet.

sekundäres Gruppenmitglied: Zusätzlich zur primären Gruppe können Benutzer in weiteren Gruppen Mitglied sein, die sekundäre Gruppenmitgliedschaft genannt wird.

normale Gruppen / Benutzer: Gruppen / Benutzer, die frei angelegt werden können und die keine systemspezifischen Aufgaben haben. Für normale Gruppen bzw. Benutzer sind Identifikationsnummern (GID bzw. UID) zwischen 100 und 65533 vorgesehen.

System-Gruppen / -Benutzer: Gruppen / Benutzer, die für spezielle System-Aufgaben zuständig sind (z.B. root, bin, named . . .). Gekennzeichnet werden diese Gruppen / Benutzer durch eine Identifikationsnummer (GID bzw. UID) unter 100 oder einer GID / UID mit dem Wert 65534.

HTTP: Hypertext-Transfer-Protokoll, zuständig für die Verständigung zwischen WWW-Server (z.B. Apache) und Webbrowser (Client, z.B. Netscape Navigator).

SSL: Secure Socket Layer, Verschlüsselungsmethode für die Verbindung zwischen WWW-Server und Webbrowser.

Perl: Practical Extraction and Report Language, eine Interpretersprache, die speziell bei Systemprogrammierern und Web-Entwicklern sehr populär ist. Ursprünglich wurde sie von Larry Wall für das Unix-Betriebssystem entworfen, heute läuft sie auf einer Vielzahl von System, z.B. OS/2, Windows, Macintosh, Plan 9.

WebMin: Webbasiertes Administrationstool für Unix-artige Betriebssysteme, welches modular aufgebaut ist. WebMin stellt einen eigenen Web-Server bereit, der standardmäßig über den Port 10000 erreichbar ist.

6.4 verwendete Literatur

- WebMin Modules, http://www.webmin.com/webmin/modules.html;
- Programmieren mit Perl; Larry Wall, Tom Christiansen und Randal L. Schwartz; O'Reilly-Verlag; Deutsche Ausgabe der 2. Auflage, 1997;
- Perl 5 kurz & gut; Johan Vromans; O'Reilly-Verlag; Deutsche Ausgabe, 2000;
- HTML kurz & gut; Jennifer Niederst; O'Reilly-Verlag; Deutsche Ausgabe, 2000;
- man-Pages zu useradd(8), usermod(8), userdel(8), groupadd(8), groupmod(8), groupdel(8), smbpasswd(8) und recode(1);

6.5 Danksagung

An dieser Stelle möchte ich mich bei der Realschule Ascheberg und Herrn Schubert bedanken, die mir dieses Projekt ermöglicht haben, bei der Firma InfoTech, Gesellschaft für Informations- und Datentechnik mbH, in Recklinghausen für die Bereitstellung des Entwicklungs-Servers sowie bei Herrn Lothar Dornieden für die Durchsicht des Pflichtenheftes. Desweiteren bei allen, die mir im Vorfeld wertvolle Anregungen zur Gestaltung und Funktionsumfang des Lösungspaketes gegeben haben und allen, die die Dokumentationen Korrektur gelesen haben.

7 Anlagen

Anlage I: Pflichtenheft

Anlage II: Benutzerhandbuch

Anlage III: Systemdokumentation

Anlage IV: Quellcode

Anlage I

Pflichtenheft

Pflichtenheft

Erstellung eines Lösungspakets zur Benutzerverwaltung auf einem Linux-Server

Tobias Brinkert
eMail: t.brinkert@15bit.de
Internet: www.15bit.de

12. Februar 2001

Inhaltsverzeichnis

1 Zielbestimmung

1.1 Mußkriterien

Das zu erstellende Lösungspaket soll sowohl die Benutzer als auch die Gruppenadministration auf dem primären Linux-Server vereinfachen. Da der Server mit Hilfe der Server-Software Samba als PDC und File-Server für die Windows-Clients dient, muß auch die Administration der Samba-Accounts gewährleistet sein. Zusätzlich soll die Möglichkeit bestehen, mithilfe einer einfachen Benutzerliste (Vorname, Nachname, Klasse), die im Dateiformat und -kodierung von Apple Macintosh-Systemen vorliegt, automatisch Accounts sowohl für den Linux- als auch für den Samba-Server zu erstellen, ein generiertes Paßwort zuweist und eine Gruppenzuweisung anhand der Klasse vornimmt. Die erzeugten Benutzerdaten sollen in eine Textdatei gespeichert werden. Gewünscht ist außerdem eine Lösung, die netzwerkfähig ist, so daß sich eine Integration in die WebMin-Oberfläche anbietet, da diese Administrationsoberfläche auf den beiden File- sowie dem Kommunikations-Servern eingesetzt und eine einheitliche Administrationsoberfläche der Server angestrebt wird.

1.2 Wunschkriterien

Wünschenswert ist die Integration des Lösungspakets in die Administrationsinfrastruktur anderer Schulen mit Linux-Servern, so daß sichergestellt werden sollte, daß die Lösung auf den unterschiedlichen Linux-Distributionen eingesetzt und an die Systemrichtlinien der einzelnen Schulen angepaßt werden kann. Zusätzlich wäre wünschenswert, daß die Benutzerliste auch im Dateiformat und -kodierung von Microsoft Windows-Systemen verarbeitet werden kann.

1.3 Abgrenzungskriterien

Das zu erstellende Lösungspaket soll die Benutzerverwaltung von WebMin nicht ersetzen, sondern nur ergänzen. Der Funktionsumfang soll auf den Bereich eingeschränkt werden, der bei den üblichen Administrationsaufgaben nötig ist, um die Wahrscheinlichkeit von Anwendungsfehlern zu minimieren.

1.4 Fertigungstellungstermin

Als Termin für die Implementierung auf dem primären File-Server der Realschule Ascheberg wird der 15. März 2001 angesetzt.

2 Produkteinsatz

2.1 Anwendungsbereich

Das Produkt soll als Ergänzung zu der bestehenden Administrationsinfrastruktur eingesetzt werden. Hierbei sollen die Funktionen der Benutzer- und Gruppenverwaltung sowie die automatische Benutzeranlegung abgedeckt werden. Ziel des Auftrages ist es, durch die Lösung die Administrationszeit zu verkürzen und die Wahrscheinlichkeit einer fehlerhaften Bedienung zu minimieren.

2.2 Zielgruppe

Die Administratoren und Informatik-Fachlehrer sind hauptsächlich für den Einsatz dieses Lösungspaketes vorgesehen.

3 Produktumgebung

3.1 Server

Für den Produkteinsatz wird eine Linux-Maschine mit dem Administrationstool WebMin ab Version 0.75 vorausgesetzt. Dabei wird vorrangig für die SuSE-Distribution Version 7.0 entwickelt. Zusätzlich muß der Zeichensatzfiler recode installiert sein, in einer Version, die die neue Syntax recode Zeichensatz1..Zeichensatz2 anstelle von recode Zeichensatz1:Zeichensatz2 interpretiert. Desweiteren wird die Scriptsprache Perl benötigt. Spezielle Anforderungen an die Hardware gibt es für den Betrieb des Produktes nicht.

3.2 Clients

Das Betriebssystem der Clients muß den Anschluß ans Netzwerk über das TCP/IP-Protokoll unterstützen, z.B. Microsoft Windows, Linux oder Apple Macintosh. Weiterhin wird ein JavaScript-fähiger Webbrowser vorausgesetzt (z.B. Microsoft Internet Explorer oder Netscape). Die Clients sollten mindestens mit einer Bildschirmauflösung von 800 x 600 Pixeln betrieben werden.

4 Produktfunktionen

Die einzelnen Produktfunktionen sind jeweils so aufgebaut, daß sie ihre Ausgaben, die Überprüfung der übergebenen Werte und die Fehlerbehandlung selber ausführen. Eine Hilfe steht in jeder Funktion zur Verfügung, wobei immer die zur Funktion gehörende Hilfeseite aufgerufen wird. Zu vielen Einstellungen, Ausgaben und Fehlermeldungen steht zusätzlich eine Kontext-Hilfe, die über Links implementiert wurde, zur Verfügung.

4.1 Produktfunktion 1: Benutzerverwaltung (F1.0)

Diese Funktion stellt die im System vorhandenen Benutzer aufgeteilt nach normalen Benutzern und System-Benutzern in Tabellenform dar. Die Werte, die für F1.1 und F1.3 nötig sind, werden über diese Funktion eingetragen, F1.2 wird mittels eines Links der Benutzername übergeben.. Die Ausführung der Funktionen F1.1, F1.2 und F1.3 wird hierüber gestartet.

4.1.1 Subfunktion 1: Benutzer anlegen (F1.1)

Die erforderlichen Eigenschaften eines neuen Benutzers werden von F1.0 an diese Funktion übergeben. Zu den Eigenschaften eines neuen Benutzers zählen der Anmelde- und realer Name, die primäre Gruppe, ob für den Benutzer ein Paßwort generiert, das Heimatverzeichnis erstellt und ein Samba-Account eingerichtet werden soll, die Zuweisung einer Shell und die Mitgliedschaft in einer primären Gruppe. Der Typ des Benutzers legt fest, wo im Verzeichnisbaum das Heimatverzeichnis angelegt werden soll. Sollte die Generierung eines Paßwortes nicht erwünscht sein, übernimmt diese Funktion die Eingabemaske für das Paßwort und die Paßwortvalidierung. Nach der Überprüfung der Eigenschaften legt diese Funktion den neuen Benutzer an oder gibt eine Fehlermeldung aus.

4.1.2 Subfunktion 2: Benutzer modifizieren (F1.2)

Diese Funktion listet Detailinformationen zu einem Benutzer auf und bietet die Möglichkeit, den Benutzer zu modifizieren. Dabei erhält es die erforderlichen Werte entweder über F1.0 (nur Benutzernamen) oder bei einer Modifizierung über sich selbst. Dabei können das Paßwort sowohl für den System- als auch für den Samba-Account verändert werden, die Zugehörigkeit zu den einzelnen Gruppen (primäre und sekundäre) modifiziert und die Eigenschaften des Samba-Accounts geändert werden.

4.1.3 Subfunktion 3: Benutzer löschen (F1.3)

Über F1.0 werden dieser Subfunktion die Benutzer übergeben, die aus dem System gelöscht werden sollen. Dabei besteht bei normalen Benutzern die Möglichkeit, das Heimatverzeichnis mit zu löschen, wobei dieses über eine Option in F1.0 angegeben wird, die standardmäßig aktiviert ist. Jede Löschaktion wird mit Statusmeldungen direkt im Browserfenster ausgegeben.

4.2 Produktfunktion 2: Gruppenverwaltung (F2.0)

Die tabellarische Darstellung der im System vorhandenen Gruppen wird von dieser Funktion übernommen. Dabei wird zwischen normalen Gruppen und System-Gruppen unterschieden. Die erforderlichen Parameter für F2.1 und F2.3 können über diese Funktion

eingetragen werden, F2.2 wird mittels eines Links der Gruppenname übergeben. Aufgerufen werden F2.1, F2.2 und F2.3 über diese Funktion.

4.2.1 Subfunktion 1: Gruppen anlegen (F2.1)

Über F2.0 erhält diese Funktion die Eigenschaften einer neuen Gruppe. Dazu zählen der Gruppenname, ob die Anlegung eines Gruppenverzeichnisses gewünscht ist und eine Auswahl, wo sich dieses dann befinden soll. Nach der Überprüfung der Eigenschaften legt diese Funktion die neue Gruppe an oder gibt eine Fehlermeldung aus.

4.2.2 Subfunktion 2: Gruppen modifizieren (F2.2)

Detailinformationen und Modifizierungen der Gruppe werden über die Funktion ausgegeben bzw. realisiert. Dabei erhält es die erforderlichen Werte entweder über F2.0 (nur Gruppenname) oder bei einer Modifizierung über sich selbst. Für die Gruppenmodifizierung steht nur die Möglichkeit offen, der Gruppe sekundäre Mitglieder hinzuzufügen oder zu entfernen.

4.2.3 Subfunktion 3: Gruppen löschen (F2.3)

Für das Entfernen von Gruppen ist diese Funktion zuständig, wobei die zu löschenden Gruppen von F2.0 übergeben werden. Jede Löschaktion wird mit Statusmeldungen direkt im Browserfenster ausgegeben.

4.3 Produktfunktion 3: Automatische Benutzeranlegung (F3.0)

Die Automatische Benutzeranlegung ist in zwei Teilen realisiert worden. Zuerst werden die notwendigen Konfigurationen abgefragt, wozu Speicherort und -format der Import- und Export-Datei, das Feldtrennzeichen, der Typ der anzulegenden Benutzer (die den Pfad zum Heimatverzeichnis festlegen) sowie ein optionales Prefix für Gruppen gehören. Nachdem diese Werte eingetragen und abgeschickt wurden, wird eine Überprüfung der Werte vorgenommen und die Benutzer angelegt, andernfalls eine Fehlermeldung ausgegeben.

4.4 Produktfunktion 4: Konfiguration (F4.0)

Gewünscht war die Integrationsfähigkeit des Produkts in die Umgebung anderer Schulen. Über diese Funktion steht die Möglichkeit offen, dieses zu realisieren. Unter anderem werden hier die verschiedenen Pfad- und Verzeichnisrechte, die Standardshell und die Spaltenanzahl für Benutzer- und Gruppentabellen festgelegt.

5 Produktdaten

Die Daten, die diese Lösung erzeugt und längerfristig benötigt werden, werden in den entsprechenden System-Datenbanken (passwd, group, shadow und smbpasswd) gespeichert. Bei der Funktion F3.0 werden die generierten Benutzerdaten samt Paßwort in eine Datei gespeichert, wobei Ort und Format frei eingestellt werden kann.

6 Benutzeroberfläche

Die Benutzeroberfläche wird für eine minimale Auflösung von 800 x 600 Pixeln optimiert, ansonsten orientiert sich das Design an die Vorgaben von WebMin (siehe [1]).

7 Qualitäts-Zielbestimmung

Ziel ist es, die Benutzer- und Gruppenadministration für Administratoren und Informatik-Fachlehrer zu vereinfachen und die erforderlichen Arbeitszeiten zu minimieren. Hierzu zählt insbesondere die leichte Einarbeitung, Verständlichkeit, Selbstbeschreibung und Zuverlässigkeit. Eine besondere Optimierung der Geschwindigkeit der automatischen Benutzeranlegung wird nicht angestrebt, da diese Funktion normalerweise nur einmal pro Schuljahr verwendet wird. Ansonsten gelten die Zielbestimmungen von WebMin (siehe [2]).

8 Globale Testszenarien

Wärend der gesamten Entwicklungszeit werden die einzelnen Funktionen auf ihre Fehlerlosigkeit überprüft. Vor der Freigabe des Lösungspakets wird die Installation auf einem weiteren System getestet und die gesamte Funktionalität mit verschiedenen Browsern und Betriebssystemen überprüft, um ein fehlerfreies Produkt zu garantieren.

9 Entwicklungsumgebung

9.1 Software

Server: SuSE-Linux-Distribution Version 7.0, WebMin Version 0.83, Perl Version 5.005_03, Samba Version 2.0.7, recode Version 3.5.

Client: SuSE-Linux-Distribution Version 7.0, Netscape Communicator Version 4.75, FTE-Editor Version 0.49.10, HTML-Editor Bluefish Version 0.3.5, WebMin Version 0.81, Perl Version 5.005_03, Samba Version 2.0.7, recode Version 3.5.

9.2 Hardware

Server: Intel Pentium-133, 80 MB RAM, 2 GB HDD, Monitor;

Client: AMD K6-500, 256 MB RAM, 15 GB + 8 GB HDD, Monitor, Drucker;

Auf dem Client wurde im globalen Testszenario das Lösungspaket zusätzlich installiert, um die Funktionalität zu prüfen.

Literatur

[1] Writing WebMin Modules, Abschnitt “Look and Feel”, http://www.webmin.com/webmin/modules.html#look

[2] Writing WebMin Modules, Abschnitt “Design Goals”, http://www.webmin.com/webmin/modules.html#design

Anlage II

Benutzerhandbuch

Benutzerhandbuch

zum WebMin-Modul:
Benutzer- und Gruppenmanager

— Version 0.2.0 —

Tobias Brinkert
eMail: t.brinkert@15bit.de
Internet: www.15bit.de

28. Februar 2001

Inhaltsverzeichnis

1 Benutzer- und Gruppenmanager

1.1 Allgemeines

Der Benutzer- und Gruppenmanager soll vielleicht nicht als Ersatz, sondern als Zusatz zum schon vorhandenen WebMin-Modul **Benutzer und Gruppen** betrachtet werden. Bei der Auswahl der Funktionalität wurde versucht, diesen Manager auf das Notwendigste zu reduzieren, um Benutzer und Gruppen zu verwalten. Andererseits wurden Funktionen der Benutzerverwaltung, die sonst über das Modul **Windows Dateifreigabe** verfügbar sind, mit in das vorliegende Modul aufgenommen, um einfach und in einem Schritt sowohl die System-Benutzer sowie die Samba-Benutzer verwalten zu können. Es soll vor allem in den Schulen Einsatz finden, wo häufig größere Benutzermengen angelegt werden müssen und Samba als Fileserver eingesetzt wird. Aus diesem Grund sind einige der Bezeichnungen auch an Begriffe aus der Schule angelehnt.

1.2 Funktionsumfang

1. **Benutzermanager**
   - Anlegen von System- und Samba-Accounts, wahlweise mit Paßworteingabe oder automatischer Paßwortgenerierung und Anlegung des Heimatverzeichnisses;
   - Anzeigen von Detailinformationen zu Accounts;
   - Ändern von Paßwörtern für den System-Account und/oder Samba-Account;
   - Ändern der Standardshell;
   - Ändern der primären Gruppe sowie der Zugehörigkeit zu sekundären Gruppen;
   - (multiples) Löschen von System- und Samba-Accounts, bei normalen Benutzern kann auch gleichzeitig das Heimatverzeichnis gelöscht werden;
2. **Gruppenmanager**
   - Anlegen von Gruppen und Gruppenverzeichnissen;
   - Anzeigen von Gruppeninformationen;
   - Gruppenmitgliedschaften für sekundäre Mitglieder ändern;
   - (multiples) Löschen von Gruppen;
3. **automatische Benutzeranlegung**
   - Anhand einer Benutzerliste (Datei) können viele Accounts automatisch eingerichtet werden;
   - Samba-Accounts und Heimatverzeichnisse anlegen;

- Paßwort automatisch generieren;
- generierte Benutzerdaten werden in eine Datei gespeichert (um den Benutzern nachher ihren Account-Namen und Paßwort mitteilen zu können);

4. **Konfiguration**
   - Für Schüler, Lehrer und Gruppen können unterschiedliche Basis-Verzeichnisse definiert werden;
   - Zuriffsrechte für neue Heimat- und Gruppenverzeichnisse können eingestellt werden, wobei hier nur einige sinnvolle Rechtevergaben zur Verfügung stehen;

1.3 Voraussetzungen

Für das Ausführen dieses Moduls müssen folgende Programme auf dem System installiert und über den Suchpfad erreichbar sein (wird beim Starten dieses Moduls überprüft):

- useradd
- userdel
- usermod
- groupadd
- groupdel
- groupmod
- chown
- mkdir
- smbpasswd
- cp
- recode (in einer Version, die die Syntax "recode ZS1..ZS2" interpretiert)

1.4 Konfiguration

Nachfolgend werden die einzelnen Parameter, die in der Konfiguration eingestellt werden können, erläutert.

- **Pfad zu Schülerheimatverzeichnissen**: Der Pfad, unter dem das Heimatverzeichnis angelegt wird, wenn als Typ "Schüler" ausgewählt wird. Desweiteren wird dieser Pfad bei der Gruppenanlegung als möglicher Ort eines Gruppenverzeichnisses verwendet.
- **Pfad zu Lehrerheimatverzeichnissen**: Der Pfad, unter dem das Heimatverzeichnis angelegt wird, wenn als Typ "Lehrer" ausgewählt wird. Desweiteren wird dieser Pfad bei der Gruppenanlegung als möglicher Ort eines Gruppenverzeichnisses verwendet.

- **Pfad zu Gruppenverzeichnissen**: Der Pfad, unter dem Gruppenverzeichnisse angelegt werden können. Bei der Gruppenanlegung gibt es 3 mögliche Orte für das Gruppenverzeichnis, wobei eines unter diesem Pfad liegt.
- **Pfad zu Profildateien (mit Shell-Benutzung)**: Die Profildateien im angegebenen Verzeichnis werden ins Heimatverzeichnis eines neuen Benutzers kopiert, wenn der Benutzer sich interaktiv am Server anmelden kann (Zuweisung der Standardshell).
- **Pfad zu Profildateien (ohne Shell-Benutzung)**: Dieser Pfad gibt das Verzeichnis mit Profildateien an, die ins Heimatverzeichnis eines neuen Benutzers kopiert werden, wenn die direkte Anmeldung am Server nicht ermöglicht wird (Shell: /bin/false).
- **Zugriffsrechte eines neuen Heimatverzeichnisses**: Hiermit kann bestimmt werden, mit welchen Zugriffsrechten das Heimatverzeichnis eines neuen Benutzers belegt werden soll. Dabei wurden einige sinnvolle Rechtekombinationen zur Auswahl gestellt um mögliche Fehlkonfigurationen auszuschließen. Generell wirde für den Benutzer ein Vollzugriff eingestellt. Darüber hinaus haben die einzelnen Auswahlmöglichkeiten folgende Bedeutung:

  700 Andere Benutzer, außer der Besitzer, haben keine Rechte auf das Verzeichnis.

  750 Mitglieder der primären Gruppe des Benutzers dürfen in das Verzeichnis hineinwechseln und den Inhalt lesen, alle anderen Benutzer nicht.

  755 Alle anderen Benutzer auf dem System dürfen in das Verzeichnis hineinwechseln und den Inhalt lesen.

  770 Mitglieder der primären Gruppe des Benutzers haben Vollzugriff auf das Verzeichnis.

  775 Mitglieder der primären Gruppe des Benutzers haben Vollzugriff auf das Verzeichnis, alle anderen Benutzer des System dürfen in das Verzeichnis hineinwechseln und den Inhalt lesen.

  777 Alle Benutzer auf dem System haben Vollzugriff auf das Verzeichnis.

- **Zugriffsrechte eines neuen Gruppenverzeichnisses**: Hiermit kann festgelegt werden, mit welchen Zugriffsrechten das Gruppenverzeichnis einer neuen Gruppe angelegt werden soll. Hierbei wurden nur drei Rechtekombinationen zur Auswahl gestellt, wobei alle Gruppenmitglieder Vollzugriff auf das Verzeichnis haben.

  770 Alle Benutzer, die nicht Mitglied der Gruppe sind, haben keinen Zugriff auf das Verzeichnis

  775 Alle Benutzer, die nicht Mitglied der Gruppe sind, dürfen in das Verzeichnis hineinwechseln und das Verzeichnis lesen.

  777 Alle Benutzer auf dem System haben Vollzugriff auf das Verzeichnis.

- **Spaltenanzahl der Benutzerlisten**: Die Spaltenanzahl der Tabellen, die im Benutzermanager aufgebaut werden.
- **Spaltenanzahl der Gruppenlisten**: Die Spaltenanzahl der Tabellen, die im Gruppenmanager aufgebaut werden.
- **Pfad/Dateiname zur Standardshell**: Sollte bei einem Benutzer die interaktive Anmeldung auf dem System ermöglicht werden, wird ihm die hier angegebene Shell zugewiesen. Bei der Modifizierung eines Benutzers wird diese Shell auch als mögliche Shell angegeben.

2 Benutzermanager

Der Benutzermanager ist für die Verwaltung aller Benutzer des Systems gedacht. Dabei werden folgende Funktionen bereitgestellt:

1. Neuen Benutzer anlegen,
2. Benutzer modifizieren und
3. Benutzer löschen.

Aufgeteilt ist die Seite in drei Teile:

1. Eingabefelder und Optionen, die für die Erstellung eines neuen Benutzers nötig sind,
2. eine Tabelle mit vorhandenen (normalen) Benutzern und
3. eine Tabelle mit vorhandenen System-Benutzern.

Ein neuer Benutzer können Sie über die Einstellungen im oberen Teil spezifizieren und über den Button **Benutzer anlegen** angelegen. Die Erläuterungen zu den einzelnen Feldern erhalten Sie in Abschnitt 2.1.1. Über den Button **Zurücksetzen** können Sie alle Formularfelder auf ihren Ursprungswert zurücksetzen.

Die nachfolgenden Tabellen zeigen alle im System vorhandenen Benutzer. Hierbei werden die Benutzer in zwei Kategorien eingeteilt: Normal und System. Über das Anklicken eines Benutzernamens gelangen Sie auf eine Seite, wo sie detailierte Benutzerinformationen erhalten sowie diese verändern können. Möchten Sie Benutzer aus dem System entfernen, markieren Sie die Checkboxen rechts neben dem Benutzernamen. Erst wenn Sie den Button **Benutzer löschen** drücken, werden die Benutzer aus dem System entfernt (über den Button **Zurücksetzen** werden alle Checkboxen demarkiert).

Für normale Benutzer kann über das Optionsfeld, welches sich unter der Tabelle für normale Benutzer befindet, ausgewählt werden, ob die Heimatverzeichnisse mit gelöscht werden sollen (siehe Abschnitt 2.1.2).

2.1 Beschreibung der Felder

2.1.1 Neuen Benutzer anlegen

Accountname: Der Anmeldename des Benutzers, wird auch für Netzwerkdienste wie Samba, FTP, Mail ... benötigt.

Realer Name: Der Vor- und Nachname des Benutzers, dem dieser Account gehört. Bei System-Accounts wird hier häufig eine Beschreibung der Funktion des Accounts angegeben.

Primäre Gruppe: Unter Unix-Betriebssystemen besteht ein Gruppenzwang, das heißt, das jeder Benutzer mindestens einer Gruppe zugeordnet werden muß. Die Hauptgruppe eines Benutzers, mit der er bei der Anmeldung initialisiert wird, wird primäre oder Initial-Gruppe genannt. Alle Dateien, die der Benutzer anlegt, werden dieser Gruppe zugeordnet. Der Benutzer kann Dateien, die ihm selbst gehören, anderen Gruppen zuordnen, falls er Mitglied der anderen Gruppe ist. Jedoch kann zu jeder Datei oder Verzeichnis nur eine Gruppe zugeordnet werden.

Paßwort automatisch generieren: Wird hier **ja** ausgewählt, generiert das System automatisch ein Paßwort aus 8 Zeichen, ansonsten erfolgt vor der Anlegung des Benutzers eine Abfrage des Paßwortes.

Typ des Benutzers: Der Typ des Benutzers gibt den Ort des Heimatverzeichnisses an. Sollte hier **Schüler** ausgewählt worden sein, wird der Pfad zum Heimatverzeichnis aus HOMEDIR_SCHUELER, primäre Gruppe und Account-Namen gebildet, für **Lehrer** aus HOMEDIR_LEHRER und Account-Namen. Die Einstellungen für HOMEDIR_SCHUELER und HOMEDIR_LEHRER können Sie in der Konfiguration zum Benutzer- und Gruppenmanager vornehmen.

Anmeldung direkt auf Server: Diese Option gibt an, ob es dem Benutzer erlaubt werden soll, sich direkt oder mittels Telnet/SSH auf dem Rechner anzumelden. Standardmäßig ist die Anmeldung direkt auf dem Server deaktiviert, da normale Benutzer die Dienste des Servers nur über diverse Netzwerkdienste benutzen sollen (z.B. Samba, Mail, FTP ...).

Samba-Benutzer einrichten: Hiermit wird festgelegt, ob für den Account auch ein Samba-Account eingerichtet und aktiviert werden soll. Standardmäßig ist dieses eingeschaltet.

Heimatverzeichnis erstellen: Für jeden Benutzer sollte ein Heimatverzeichnis auf dem Server bestehen. Sollte es Gründe geben, daß für den Benutzer kein neues Heimatverzeichnis erstellt werden soll, kann dieses Feature hier deaktiviert werden. Soll ein neues Heimatverzeichnis erstellt werden, wird ein Grundprofil in das Heimatverzeichnis kopiert. Der Ort dieses Grundprofils wird in der Konfiguration über "Pfad

zu Profildateien" angegeben. Je nach Einstellung von "Anmeldung direkt auf Server" werden unterschiedliche Profildateien kopiert.

2.1.2 Benutzer löschen

Heimatverzeichnisse löschen: Beim Löschen normaler Benutzer können die Heimatverzeichnisse mit gelöscht werden. Da das normalerweise sinnvoll ist, ist diese Eigenschaft standardmäßig aktiviert. Möchten Sie mehrere Benutzer löschen, wobei bei einigen das Heimatverzeichnis nicht gelöscht werden soll, müssen Sie dies in zwei Schritten tun, eine expliziete Auswahl, von wem die Heimatverzeichnisse gelöscht werden, ist nicht vorgesehen.

Sollten Sie System-Benutzer löschen, wird hier **nicht** automatisch das Heimatverzeichnis gelöscht, da sich diese teilweise im normalen Dateisystem befinden und nicht gelöscht werden können, ohne das wichtige Teile des Betriebssystems beschädigt würden. Um mögliche Systemausfälle zu vermeiden, wurde deshalb dieses Feature nicht implementiert.

3 Benutzer anlegen

Sollten Sie bei der Eingabe der Benutzerdaten die automatische Paßwortgenerierung abgeschaltet haben, werden Sie nach einem Paßwort gefragt und erst nach dem Abschicken des Formulars wird der Benutzer angelegt.

Nach der Überprüfung der eingegebenen Werte wird der Benutzer angelegt und die Benutzereinstellungen ausgegeben. Wird das Paßwort automatisch generiert, wird dieses hier mit ausgegeben.

Bei Fehlern in der Eingabe wird eine entsprechende Warnmeldung ausgegeben und der Benutzer nicht angelegt. Über den Zurück-Button des Browsers können Sie auf die vorherige Seite zurückwechseln und nach Veränderung der Benutzereinstellungen den Vorgang des Anlegens eines neuen Benutzers neu initiieren.

Bitte beachten Sie, daß das Heimatverzeichnis **nicht** vorhanden sein darf wenn es neu angelegt werden soll, ansonsten kommt es zu einer Fehlermeldung. Das Heimatverzeichnis wird für Schüler in HOMEDIR_SCHULER/Gruppe/Account-Name angelegt, für Lehrer in HOMEDIR_LEHRER/Account-Name. Die Einstellungen für HOMEDIR_SCHUELER und HOMEDIR_LEHRER können Sie in der Konfiguration zum Modul **Benutzer- und Gruppenmanager** vornehmen.

4 Benutzer modifizieren

Dieses Modul zeigt Detail-Informationen zu einem Account und ermöglicht die Modifizierung einiger Einstellungen für den Benutzer. Dabei gliedert sich diese Seite in 3 Teile:

1. **Rückmeldung der Benutzermodifizierung**
   Hier wird der Status der Benutzermodifizierung mit eventuellen Fehlermeldungen ausgegeben. Dieser Teil wird nur angezeigt, wenn die Benutzerdaten modifiziert wurden.

2. **Benutzer-Details für ausgewählten Benutzer**
   Neben dem Account und realen Namen werden die wohl wichtigsten Daten zum Benutzer angezeigt. Das ist der Pfad zum Heimatverzeichnis, die zugewiesende Shell, die UID sowie die primäre Gruppe und die Mitgliedschaften in sekundären Gruppen. Bei der primären und den sekundären Gruppen wird in Klammern die Gruppen-Identifikationsnummer (GID) mit ausgegeben.

3. **Benutzer-Daten ändern für ausgewählten Benutzer**
   Im dritten Teil haben Sie die Möglichkeit, einige der Benutzer-Einstellungen zu verändern. Dabei habe ich mich auf die wichtigsten Einstellungen beschränkt, die wahrscheinlich im Schulalltag verwendet werden. Dazu gehören die Paßwortänderung und die Zuweisung zu sekundären Gruppen. Sind weitergehende Einstellungsänderungen notwendig, die in diesem WebMin-Modul (noch) nicht implementiert sind, bitte ich Sie, das Modul **"Benutzer und Gruppen"** , welches mit WebMin mitgeliefert wird, zu verwenden. Die Unterstützung von Quota ist auch noch nicht implementiert, hier ist das Modul **"Festplatten-Quota"** zu verwenden.

   Alle Werte des Formulars sind standardmäßig auf die **aktuellen Benutzerwerte** bzw. auf **keine Änderung** gesetzt. Sollte der Button **Benutzer ändern** gedrückt werden, ohne daß Veränderungen in den Benutzer-Daten vorgenommen wurden, werden keine Änderungen durchgeführt. Der Button **Zurücksetzen** setzt die Werte auf die aktuellen Benutzerdaten zurück.

   Bitte beachten Sie, daß die Gruppenzugehörigkeiten von Dateien und Verzeichnissen (vor allem im Heimatverzeichnis) **nicht** verändert werden sowie das Heimatverzeichnis nicht verschoben wird.

4.1 Beschreibung der einzelnen Felder

4.1.1 Benutzer-Detail

Accountname: Der Anmeldename des Benutzers, wird auch für Netzwerkdienste wie Samba, FTP, Mail ... benötigt.

Benutzer ID (UID): Die Benutzer ID (UID) ist eine eindeutige Nummer im System, die den Benutzer identifiziert. Intern werden die Besitzer von Dateien und Verzeichnissen nicht mit ihrem Namen sondern mit der UID gespeichert. Hieraus ergibt sich, daß Dateien, die von einem User angelegt wurden und der User anschließend gelöscht wird, immer noch der UID des anlegenden Benutzers gehören. Wird ein neuer User mit der UID des gelöschten Users angelegt, gehören ihm alle Dateien des gelöschten Benutzers.

UIDs unter 100 sind typischerweise für spezielle System-Benutzer reserviert (z.B. bin, mysql, named ...). Der Benutzer mit der UID 0 ist immer der Systemadministrator (root), der mit der UID 65534 immer "nobody". Insgesamt stehen 65535 UIDs für Accounts zur Verfügung, für normale Accounts bleiben 65434 UIDs.

Realer Name: Der Vor- und Nachname des Benutzers, dem dieser Account gehört. Bei System-Accounts wird hier häufig eine Beschreibung der Funktion des Accounts angegeben.

Primäre Gruppe: Unter Unix-Betriebssystemen besteht ein Gruppenzwang, das heißt, daß jeder Benutzer mindestens einer Gruppe zugeordnet werden muß. Die Hauptgruppe eines Benutzers, womit er bei der Anmeldung initialisiert wird, wird primäre oder Initial-Gruppe genannt. Alle Dateien, die der Benutzer anlegt, werden dieser Gruppe zugeordnet. Der Benutzer kann Dateien, die ihm selbst gehören, anderen Gruppen zuordnen, falls er Mitglied der anderen Gruppe ist, jedoch kann zu jeder Datei oder Verzeichnis nur eine Gruppe zugeordnet werden.

sekundäre Gruppen: Die Gruppen, in denen der Benutzer zusätzlich zur primären Gruppe Mitglied ist.

Heimatverzeichnis: Der Pfad zum Heimatverzeichnis des Benutzers.

Shell: Die Shell, die dem Benutzer für die Anmeldung zugewiesen wurde. Reale Shells (z.B. /bin/bash) sind nur für Benutzer notwendig, die sich direkt oder per Telnet / SSH an dieser Maschine anmelden sollen. Für Benutzer, die nur einen Account auf dieser Maschine benötigen, um Netzwerkdienste zu benutzen (z.B. Samba, Mail, FTP ...), sollte die Shell **"/bin/false"** zugewiesen werden, da damit eine direkte Anmeldung oder eine Anmeldung per Telnet/ SSH nicht ermöglicht wird.

4.1.2 Benutzer-Daten ändern

Paßwort ändern: Über diesen Punkt kann erreicht werden, daß das System-Paßwort geändert wird (nicht das Samba-Paßwort). Um unterschiedliche Paßwörter zu vermeiden, sollte bei Änderung des System-Paßwort auch das Samba-Paßwort geändert werden (siehe Samba-Account). Das neue Paßwort wird über die Felder Paßwort und Paßwortwiederholung angegeben.

Paßwort und Paßwortwiederholung: Hier kann das neue Paßwort des Benutzers eingegeben werden, das maximal 14 Zeichen betragen kann. Zur Validierung muß im Feld Paßwortwiederholung das gleiche Paßwort eingetragen werden. Das neue Paßwort wird nur gesetzt, wenn die beiden angegebenen Paßwörter identisch sind sowie Paßwort ändern auf **"ja"** und/oder Samba-Account auf **"Paßwort ändern"** oder **"anlegen und aktivieren"** gesetzt sind.

Primäre Gruppe: Hier kann die primäre Gruppe des Benutzers verändert werden. Standardmäßig steht hier die aktuelle primäre Gruppe. Sollten Sie die primäre Gruppe ändern, achten Sie bitte darauf, daß diese im System vorhanden sein muß. Über den Button neben dem Eingabefeld erhalten Sie ein Fenster mit allen im System vorhandenen Gruppen. Klicken Sie eine der Gruppen an, wird dieser Gruppenname in das Feld für die primäre Gruppe übernommen.

Bitte beachten Sie, daß die Gruppenzugehörigkeiten von Dateien und Verzeichnissen (vor allem im Heimatverzeichnis) **nicht** verändert werden sowie das Heimatverzeichnis nicht verschoben wird.

aus sekundären Gruppen löschen: Die Liste enthält alle Gruppen mit der GID in Klammern, in denen der Benutzer sekundäres Mitglied ist. Möchten Sie den Benutzer aus den sekundären Gruppen löschen, markieren Sie bitte diese Gruppen. Eine Mehrfachauswahl ist möglich.

Bitte beachten Sie, daß, wenn sie sehr viele Gruppen markieren, es vorkommen kann, daß der Benutzer nicht aus allen sekundären Gruppen gelöscht wird. Wenn dieser Fall auftritt, wiederholen sie bitte die Modifizierung der Zugehörigkeit zu den sekundären Gruppen.

zu sekundären Gruppen hinzufügen: Die Liste enthält alle Gruppen mit der GID in Klammern, in denen der Benutzer sekundäres Mitglied werden kann. Möchten Sie den Benutzer zu weiteren sekundären Gruppen hinzufügen, markieren Sie bitte die entsprechenden Gruppen. Eine Mehrfachauswahl ist möglich.

Bitte beachten Sie, daß, wenn sie sehr viele Gruppen markieren, es vorkommen kann, daß der Benutzer nicht zu allen sekundären Gruppen hinzugefügt wird. Wenn dieser Fall auftritt, wiederholen sie bitte die Modifizierung der Zugehörigkeit zu den sekundären Gruppen.

Samba-Account: Die Eigenschaften des Samba-Accounts können für den aktuellen Benutzer hier verändert werden. Dabei stehen die folgenden Möglichkeiten zur Verfügung:

keine Änderung: Es werden keine Änderungen am Samba-Account vorgenommen.

***anlegen und aktivieren*:** Für den Benutzer wird ein neuer Samba-Account mit dem angegebenen Paßwort eingerichtet. Sollte der Benutzer schon vorhanden sein, wird der Account aktiviert und auf das neue Paßwort gesetzt.

aktivieren: Hiermit kann ein gesperrter Samba-Account wieder aktiviert werden. Account-Name und Paßwort bleiben erhalten.

deaktivieren: Den Samba-Account des Benutzers sperren. Dabei bleibt der Account-Name und das Paßwort erhalten, Authentifizierungen gegenüber dem Samba-Server werden zurückgewiesen.

Paßwort ändern: Das Paßwort für den Samba-Account ändern (siehe Paßwort und Paßwortwiederholung).

löschen: Löschen des Samba-Accounts, der Account muß vorhanden sein.

Shell: Das Dropdown-Feld zeigt die Shells, die dem Benutzer zugewiesen werden können. Standardmäßig enthält die Liste nur zwei Shells: **“/bin/false”** und die in der Konfiguration des Benutzer- und Gruppenmanagers angegebene **“Standardshell”** . Sollte der Benutzer derzeit eine andere Shell benutzen, wird diese Shell der Liste hinzugefügt.

5 Benutzer löschen

Die Benutzer, die im Benutzermanager zum Löschen markiert wurden, werden jetzt gelöscht. Hierbei wird für jeden Benutzer der Status der Accountentfernung angezeigt. Der Status setzt sich aus drei Werten zusammen:

1. **aus System löschen** [erfolgt | gescheitert]

   - **erfolgt** kennzeichnet eine erfolgreiche Ausführung und
   - **gescheitert** kennzeichnet eine Ausführung mit Fehlern. Der Fehlercode wird in Klammern mit angegeben, über den Link können Sie nähere Informationen über die Art des Fehlers erhalten. Bei einem Fehler 12 konnte das Heimatverzeichnis des Benutzers nicht gelöscht werden, da es nicht mehr vorhanden ist (z.B. durch vorheriges Löschen oder Verschieben). Der System-Account wurde jedoch trotzdem gelöscht und steht nicht mehr zur Verfügung.

2. **Heimatverzeichnis löschen** [erfolgt | ausgelassen]

   - Bei **erfolgt** wurde das Heimatverzeichnis des Benutzers erfolgreich gelöscht.
   - In den Fällen, das die Option “Heimatverzeichnis löschen” nicht eingeschaltet ist, es sich bei dem zu löschenden Account um einen Systemaccount handelt oder das Heimatverzeichnis nicht mehr vorhanden ist, wird der Status **ausgelassen** zurückgegeben.

3. **aus Samba löschen** [erfolgt | nicht vorhanden]

- Bei der Rückmeldung **erfolgt** konnte der Samba-Account erfolgreich aus dem System entfernt werden,
- die Meldung **nicht vorhanden** kennzeichnet einen gescheiterten Versuch. Die Ursache hierfür liegt fast immer darin, daß es keinen Samba-Account für den Benutzer gibt. Daß der Fehler andere Ursachen haben könnte, kann normalerweise ausgeschlossen werden.

Das Heimatverzeichnis eines Accounts wird nur gelöscht, wenn im Benutzermanager das entsprechende Optionsfeld eingeschaltet ist (Standard) und es sich nicht um einen System-Account handelt. Desweiteren versucht dieses Modul auf jedem Fall, den Samba-Account zu löschen, da eine Anmeldung über Samba ohne entsprechenden Unix/Linux im Normalfall nicht möglich ist und somit Inkonsistenzen in den Benutzerdatenbanken vermieden werden können.

Sollten Fehler beim Löschen von Accounts auftreten, wird eine entsprechende Meldung mit der Gesamtzahl der Fehler ausgegeben. Zum Abschluß werden noch Laufzeit-Daten über die Scriptausführung wiedergegeben.

6 Gruppenmanager

Der Gruppenmanager ist für die Verwaltung aller Gruppen des System gedacht. Dabei werden folgende Funktionen bereitgestellt:

1. Neue Gruppe anlegen,
2. Gruppe modifizieren und
3. Gruppe(n) löschen.

Aufgeteilt ist die Seite in drei Teile:

1. Eingabefeld und Optionen, die für die Erstellung einer neuen Gruppe nötig sind,
2. eine Tabelle mit vorhandenen (normalen) Gruppen und
3. eine Tabelle mit vorhandenen System-Gruppen.

Eine neue Gruppe können Sie über die Einstellungen im oberen Teil spezifizieren und über den Button **Gruppe anlegen** angelegen. Die Erläuterungen zu den einzelnen Feldern erhalten sie im Abschnitt 6.1.1. Über den Button **Zurücksetzen** können Sie alle Formularfelder auf ihren Ursprungswert zurücksetzen.

Die nachfolgenden Tabellen zeigen alle im System vorhandenen Gruppen. Hierbei werden die Benutzer in zwei Kategorien eingeteilt: Normal und System. Die Funktionalität

ist genauso wie im Benutzermanager implementiert worden. Über das Anklicken eines Gruppennamens gelangen Sie auf eine Seite, wo sie detailierte Gruppeninformationen erhalten sowie diese verändern können. Möchten Sie Gruppen aus dem System entfernen, markieren Sie die Checkboxen rechts neben dem Gruppennamen. Erst wenn Sie den Button **Gruppen löschen** drücken, werden die Gruppen aus dem System entfernt (über den Button **Zurücksetzen** werden alle Checkboxen demarkiert).

Derzeit besteht noch nicht die Möglichkeit, eventuell vorhandene Gruppenverzeichnisse zu löschen.

6.1 Beschreibung der Felder

6.1.1 Neue Gruppe anlegen

Gruppenname: Der Name, unter der die Gruppe im System bekannt gegeben werden soll.

Gruppenverzeichnis anlegen: Hier ist die Möglichkeit gegeben, ein Gruppenverzeichnis anzulegen. Ist das Kontrollkästchen aktiviert (Standard), wird ein Gruppenverzeichnis angelegt. Der Pfad wird über das Drop-Down Menü Gruppenverzeichnis angegeben.

Gruppenverzeichnis: Für den Ort des Gruppenverzeichnisses stehen 3 Möglichkeiten offen:

1. Pfad zu Gruppenverzeichnissen + Gruppenname
2. Pfad zu Schülerheimatverzeichnissen + Gruppenname + Gruppenname (Standard)
3. Pfad zu Lehrerheimatverzeichnissen + Gruppenname

Dabei werden die Zugriffsrechte des Gruppenverzeichnisses auf den Wert eingestellt, der in der Konfiguration zum **Benutzer- und Gruppenmanager** eingetragen wurden. Das Gruppenverzeichnis darf nicht existieren.

7 Gruppe anlegen

Nach der Überprüfung der eingegebenen Werte wird die Gruppe angelegt und die Gruppeneinstellungen ausgegeben.

Bei Fehlern in der Eingabe wird eine entsprechende Warnmeldung ausgegeben und die Gruppe nicht angelegt. Über den Zurück-Button des Browsers können Sie auf die vorherige Seite zurückwechseln und nach Veränderung der Gruppeneinstellungen den Vorgang des Anlegens einer neuen Gruppe neu initiieren.

Bitte beachten Sie, daß das Gruppenverzeichnis **nicht** vorhanden sein darf wenn es neu angelegt werden soll, ansonsten kommt es zu einer Fehlermeldung.

8 Gruppe modifizieren

Dieses Modul zeigt Detail-Informationen zu einer Gruppe und ermöglicht die Modifizierung der sekundären Mitglieder der Gruppe. Dabei gliedert sich diese Seite in 3 Teile:

1. **Rückmeldung der Gruppenmodifizierung**

   Hier wird der Status der Gruppenmodifizierung mit eventuellen Fehlermeldungen ausgegeben. Dieser Teil wird nur angezeigt, wenn die Gruppendaten modifiziert werden.

2. **Gruppen-Details für ausgewählte Gruppe**

   Neben dem Gruppennamen und der Gruppen ID (GID) werden die primären und sekundären Gruppenmitglieder ausgegeben. Bei den Gruppenmitgliedern wird sowohl der Name als auch in Klammern die UID angezeigt..

3. **Gruppen-Daten ändern für ausgewählte Gruppe**

   Im dritten Teil haben Sie die Möglichkeit, der Gruppe sekundäre Mitglieder hinzuzufügen oder zu entfernen. Die Unterstützung von Quota ist (noch) nicht implementiert, hier ist das Modul **"Festplatten-Quota"** zu verwenden.

   Alle Werte des Formulars sind standardmäßig auf die **aktuellen Gruppenwerte** gesetzt. Sollte der Button **Gruppe ändern** gedrückt werden, ohne daß Veränderungen in den sekundären Mitgliedern der Gruppe vorgenommen wurden, werden keine Änderungen durchgeführt. Der Button **Zurücksetzen** setzt die Werte auf die aktuellen Werte zurück.

8.1 Beschreibung der einzelnen Felder

8.1.1 Gruppen-Detail

Gruppenname: Der Gruppenname, wie er im System verwendet wird.

Gruppen ID (GID): Die Gruppen ID (GID) ist eine eindeutige Nummer im System, die die Gruppe identifiziert. Intern werden die Gruppenzugehörigkeiten von Dateien und Verzeichnissen nicht mit ihrem Namen sondern mit der GID gespeichert. Hieraus ergibt sich, daß Dateien, die einer Gruppe gehören und die Gruppe anschließend gelöscht wird, immer noch der GID der Gruppe gehören. Wird eine neue Gruppe mit der GID der gelöschten Gruppe angelegt, gehören alle Dateien und Verzeichnisse der gelöschten Gruppe der neuen Gruppe.

GIDs unter 100 sind typischerweise für spezielle System-Gruppen reserviert (z.B. bin, modem, named ...). Die Gruppe mit der GID 0 ist immer die Systemadministratorgruppe (root), die mit der GID 65534 ist immer "nogroup". Insgesamt stehen 65535 GIDs für Gruppen zur Verfügung, für normale Gruppen bleiben 65434 UIDs.

primäre Gruppenmitglieder: Unter Unix-Betriebssystemen besteht ein Gruppenzwang, d.h., das jeder Benutzer mindestens einer Gruppe zugeordnet werden muß. Die Hauptgruppe eines Benutzers, womit er bei der Anmeldung initialisiert wird, wird primäre oder Initial-Gruppe genannt. Alle Dateien, die der Benutzer anlegt, werden dieser Gruppe zugeordnet. Natürlich können mehrere Benutzer ein und die selbe Gruppe als primäre Gruppe nutzen.

Primäre Gruppenmitglieder können mit diesem Modul nicht verändert werden, dazu müssen Sie die Benutzereinstellungen im Benutzermanager modifizieren.

sekundäre Gruppenmitglieder: Die Benutzer, die in dieser Gruppe zusätzlich zur ihrer primären Gruppe als Mitglied eingetragen sind. Hiermit können zusätzlich Gruppen zu den Hauptgruppen gebildet werden, die gemeinsame Rechte besitzen.

8.1.2 Sekundäre Gruppenmitglieder ändern

Gruppenmitglied(er) entfernen: Die Liste enthält alle Benutzer mit der UID in Klammern, die in dieser Gruppe sekundäres Mitglied sind. Möchten Sie sekundäre Mitglieder aus dieser Gruppen löschen, markieren Sie bitte diese Benutzer. Eine Mehrfachauswahl ist möglich.

Bitte beachten Sie, daß, wenn sie sehr viele Mitglieder markieren, es vorkommen kann, daß nicht alle sekundären Mitglieder der Gruppen gelöscht werden. Wenn dieser Fall auftritt, wiederholen sie bitte die Modifizierung der Zugehörigkeit zu der Gruppe.

Gruppenmitglied(er) hinzufügen: Die Liste enthält alle Benutzer mit der UID in Klammern, die in dieser Gruppe sekundäres Mitglied werden können. Möchten Sie der Gruppe weiteren sekundären Mitglieder hinzufügen, markieren Sie bitte die entsprechenden Benutzer. Eine Mehrfachauswahl ist möglich.

Bitte beachten Sie, daß, wenn sie sehr viele Benutzer markieren, es vorkommen kann, daß nicht alle der Gruppe als sekundäres Mitglieder hinzugefügt werden. Wenn dieser Fall auftritt, wiederholen sie bitte die Modifizierung der Zugehörigkeit zu der Gruppe.

9 Gruppe löschen

Die Gruppen, die im Gruppenmanager zum Löschen markiert wurden, werden jetzt gelöscht. Hierbei wird für jede Gruppe der Status der Gruppenentfernung angezeigt. Die

möglichen Rückgabewerte sind:

1. **erfolgt** wird für eine erfolgreiche Ausführung verwendet und
2. **gescheitert** für eine Ausführung mit Fehlern. Der Fehlercode wird in Klammern mit angegeben, über den Link können Sie nähere Informationen über die Art des Fehlers erhalten.

Sollten Fehler beim Löschen von Gruppen auftreten, wird eine entsprechende Meldung mit der Gesamtzahl der Fehler ausgegeben. Zum Abschluß werden noch Laufzeit-Daten über die Scriptausführung wiedergegeben.

10 Mehrere Benutzer anlegen

Dieses Modul ermöglicht es, schnell und einfach viele Benutzer auf einem Server mit Hilfe einer Listendatei anzulegen. Hierfür muß die Listendatei folgende Daten für jeden anzulegenden Benutzer enthalten:

1. Vorname
2. Nachname
3. Gruppe

Diese drei Eigenschaften müssen jeweils mit einem Trennzeichen in einer Zeile stehen. Mögliche Trennzeichen sind **“,”** , **“;”** , **“#”** und **“:”** . Dabei müssen die Trennzeichen innerhalb der Datei immer gleich sein und es darf pro Zeile nur ein Datensatz enthalten sein (auch keine Kommentare oder ähnliches). Die ganze Datei darf keine weiteren Daten enthalten.

Da die Benutzerdaten von einem anderen Betriebssystem als Linux stammen könnten und die weitere Verarbeitung der generierten Benutzerdaten auch auf einem anderen Betriebssystem stattfinden kann, muß das Ursprungs- und Zielbetriebssystem für die beiden Dateien angegeben werden, da die Textdateien unter den verschiedenen Betriebssystemen andere Zeichensätze verwenden.

Wird ein Prefix für Gruppen angegeben, wird der Gruppe des Benutzers, die in der Listendatei eingetragen ist, dieses Prefix vorangestellt (s.u.). Alle Gruppen, die den Benutzern zugewiesen werden sollen, müssen vorher im System vorhanden sein, ansonsten wird eine Fehlermeldung mit den noch fehlenden Gruppen ausgegeben und die Ausführung beendet.

Der Accountname wird anhand des Vor- und Nachnamens gebildet und komplett in Kleinbuchstaben umgewandelt. Hierfür werden zuerst die Sonderzeichen, die in einem Namen vorkommen können, entfernt und die Umlaute ersetzt. Danach wird versucht einen Anmeldenamen zu generieren, der aus dem ersten Buchstaben des Vornamens und dem

Nachnamen zusammengesetzt wird, jedoch höchstens 12 Zeichen. Ist der Name im System schon vorhanden, werden die ersten beiden Buchstaben des Vornamens und der Nachname genommen usw. Sollte kein Anmeldename generiert werden können, wird der Benutzer nicht angelegt und mit dem nächsten fortgefahren (nach entsprechender Fehlermeldung).

Für jeden Benutzer in der Listendatei werden folgende Einstellungen durchgeführt:

1. System-Account einrichten
2. Paßwort generieren (8 Zeichen), wobei das Paßwort aus dem mit Hilfe des Soundex-Algorithmus gepackten Anmeldenamen und 4 weiteren zufälligen Zeichen besteht. Sollte der Soundex-Algorithmus keinen Wert liefern können, wird das Paßwort komplett aus 8 zufälligen Zeichen generiert.
3. Profildateien ins Heimatverzeichnis kopieren
4. Samba-Account einrichten und aktivieren

Die folgenden Daten werden für jeden Benutzer in die Ausgabedatei geschrieben:

1. Primäre Gruppe
2. Vor- und Nachname
3. Anmeldename
4. generiertes Paßwort (8 Zeichen)
5. Heimatverzeichnis
6. Status der Benutzeranlegung

10.1 Beschreibung der einzelnen Felder

Datei mit den Benutzerdaten (mit Pfad): Der Dateiname mit den Benutzern, die angelegt werden sollen. Hierbei muß der komplette Pfad angegeben werden. Der Button neben dem Eingabefeld ermöglicht das Navigieren im Dateisystem und das Auswählen einer Datei.

Pfad für die Datei mit den generierten Benutzerdaten: Für die Ausgabedatei mit den generierten Benutzerdaten samt Paßwörtern wird hier der Pfad abgelegt, wo diese Datei gespeichert werden soll. Der Name dieser Datei lautet **userlist_<Zahl>.txt**, wobei die <Zahl> die Prozeß-ID ist. Bitte beachten Sie, daß eine gleichnamige Datei in dem Verzeichnis kommentarlos überschrieben wird. Da diese Datei die Paßwörter der angelegten Benutzer enthält, sollte diese nicht längerfristig auf dem Server gespeichert werden.

Typ der Benutzer: Diese Auswahl gibt den Ort des Heimatverzeichnisses der anzulegenden Benutzer an. Wird **Schüler** gewählt, werden der Pfad zum Heimatverzeichnis aus HOMEDIR_SCHUELER + Gruppe + generiertem Account-Name gebildet, bei **Lehrer** aus HOMEDIR_LEHRER + generiertem Account-Namen. Die Einstellungen für HOMEDIR_SCHUELER und HOMEDIR_LEHRER können über das Konfigurationsmenü von **Benutzer- und Gruppenverwaltung** angepaßt werden.

Trennzeichen in Benutzerdaten: Legt das Trennzeichen für die einzelnen Spalten der Benutzerdaten fest. Das Trennzeichen gilt sowohl für die Datei mit den Benutzerdaten als auch für die generierten Benutzerdaten.

Format der Benutzerdaten und Format der generierten Benutzerdaten: Die beiden Dropdown Felder legen das System fest, von dem die Benutzerdaten stammen und mit welchem sie weiterverarbeitet werden sollen.

Prefix für Gruppen: Dieser String wird der in den Benutzerdaten angegebenen Gruppe vorangestellt und daraus ein Gruppenname generiert. Wird hier ein String angegeben und ist in den Benutzerdaten ein Gruppenname angegeben, der nur aus einer Ziffer und einem Buchstaben besteht, wird hier der Ziffer eine **0** vorangestellt, so daß eine zweistellige Ziffer generiert wird.

11 Status der automatischen Benutzeranlegung

Das Programm zur automatischen Anlegung von Benutzern gibt während der gesamten Laufzeit Statusinformationen über den Stand der Arbeit sowie einige Übersichten zum Abschluß aus. Zuerst wird geprüft, ob alle benötigten Gruppen vorhanden sind und über diese Prüfung eine Meldung ausgegeben. Während des Anlegens der einzelnen Benutzer wird der Name des aktuell bearbeiteten Benutzers, der generierte Account-Name, die zugeordnete primäre Gruppe sowie eine Bearbeitungsmeldung ausgegeben.

Nachdem alle Benutzer angelegt wurden, wird eine Gruppenübersicht erstellt mit allen Gruppen und wieviele Benutzer diesen Gruppen zugeordnet wurden. Anschließend wird die Gesamtzahl der angelegten Accounts und Samba-Accounts aufgelistet. Bei einer komplett erfolgreichen Anlegung sollten diese beiden Zahlen gleich sein.

Zum Abschluß wird noch der Pfad und Dateiname zur Datei mit den generierten Benutzerdaten sowie Laufzeitdaten ausgegeben.

12 Rückgabecodes von Systemprogrammen

Programm "useradd"

RC = 0 Benutzer erfolgreich eingerichtet.

RC = 1 Kann die Paßwort-Datei nicht aktualisieren.

RC = 2 Die Kommandosyntax ist fehlerhaft.

RC = 3 Fehlerhafte Argumente für Optionen.

RC = 4 UID ist nicht eindeutig.

RC = 6 Angegebene Gruppe existiert nicht.

RC = 9 Benutzername ist nicht eindeutig.

RC = 10 Kann die Gruppendatei nicht aktualisieren.

RC = 12 Kann das Heimatverzeichnis nicht anlegen.

Programm "usermod"

RC = 0 Benutzer erfolgreich modifiziert.

RC = 1 Kann die Paßwort-Datei nicht aktualisieren.

RC = 2 Kommandosyntax fehlerhaft.

RC = 3 Fehlerhafte Argumente für Optionen.

RC = 4 UID ist nicht eindeutig.

RC = 5 Paßwort-Datei enthält Fehler.

RC = 6 Angegebener Benutzer/Gruppe existiert nicht.

RC = 8 Der Benutzer ist zur Zeit angemeldet.

RC = 9 Benutzername ist nicht eindeutig.

RC = 10 Kann die Gruppendatei nicht aktualisieren.

Programm "userdel"

RC = 0 Benutzer erfolgreich gelöscht.

RC = 1 Kann die Paßwort-Datei nicht aktualisieren.

RC = 2 Kommandosyntax fehlerhaft.

RC = 6 Angegebener Benutzer existiert nicht.

RC = 8 Der Benutzer ist zur Zeit auf dem System angemeldet.

RC = 10 Kann die Gruppendatei nicht aktualisieren.

RC = 12 Kann das Heimatverzeichnis nicht löschen.

Programm “groupadd”

RC = 0 Gruppe erfolgreich angelegt.

RC = 2 Kommandosyntax fehlerhaft.

RC = 3 Fehlerhafte Argumente für Optionen.

RC = 4 GID ist nicht eindeutig.

RC = 9 Gruppenname ist nicht eindeutig.

RC = 10 Kann die Gruppendatei nicht aktualisieren.

Programm “groupmod”

RC = 0 Gruppe erfolgreich modifiziert.

RC = 2 Kommandosyntax fehlerhaft.

RC = 3 Fehlerhafte Argumente für Optionen.

RC = 4 GID ist nicht eindeutig.

RC = 6 Angegebene Gruppe existiert nicht.

RC = 9 Gruppenname ist nicht eundeutig.

RC = 10 Kann die Gruppendatei nicht aktualisieren.

Programm “groupdel”

RC = 0 Gruppe erfolgreich gelöscht.

RC = 2 Kommandosyntax fehlerhaft.

RC = 6 Angegebene Gruppe existiert nicht.

RC = 8 Kann die primäre Gruppe eines Benutzers nicht löschen.

RC = 10 Kann die Gruppendatei nicht aktualisieren.

Programm “smbpasswd”

RC = 1 Beim Verändern eines Samba-Accounts (Befehl “smbpasswd”) ist ein Fehler aufgetreten.

Programm “chown”

RC = 1 Beim Ändern der Benutzer- und Gruppenzugehörigkeit (Befehl “chown”) ist ein Fehler aufgetreten.

Programm “mkdir”

RC = 1 Beim Erstellen eines neuen Verzeichnisses (Befehl “mkdir”) ist ein Fehler aufgetreten.

Programm “rm”

RC = 1 Beim Löschen (Befehl “rm”) ist ein Fehler aufgetreten.

Programm “cp”

RC = 1 Bei dem Kopieren (Befehl “cp”) ist ein Fehler aufgetreten.

Programm “recode”

RC = 1 Bei der Ausführung des Programms “recode” ist ein Fehler aufgetreten. Wenn Sie dieses Modul das erste Mal benutzen, überprüfen Sie bitte, ob ihr Programm “recode” die neue Syntax verwendet (recode ZS1..ZS2 anstelle von recode ZS1:ZS2).

Anlage III

Systemdokumentation

Systemdokumentation

zum WebMin-Modul:
Benutzer- und Gruppenmanager

— Version 0.2.0 —

Tobias Brinkert
eMail: t.brinkert@15bit.de
Internet: www.15bit.de

04. März 2001

Inhaltsverzeichnis

1 Beschreibung der einzelnen Scripte[1]

1.1 Script: index.cgi

Funktion: Startseite für Benutzer/Gruppenmanager generieren, Prüfung der Konfigurationseinstellungen;

verknüpfte Scripte: user.cgi, group.cgi, userauto.cgi, config.cgi*, help.cgi*

1.2 Script: user.cgi

Funktion: Zeigt alle vorhandenen Benutzer, Felder für die Benutzeranlegung, Felder für die Benutzerlöschung, Link für die Benutzermodifizierung

verknüpfte Scripte: useradd.cgi, usermod.cgi, userdel.cgi, index.cgi, help.cgi*

1.3 Script: useradd.cgi

Funktion: Legt neuen Benutzer an

verknüpfte Scripte: user.cgi, index.cgi, help.cgi*

1.3.1 Funktion: passwdpart()

Übergabewert: keine

Rückgabewert: keine

Beschreibung: Liefert HTML-Code für die Eingabe eines Paßwortes

1.4 Script: usermod.cgi

Funktion: Zeigt Benutzer-Details und modifiziert Benutzerdaten

verknüpfte Scripte: usermod.cgi, user.cgi, index.cgi, help.cgi*

1.5 Script: userdel.cgi

Funktion: Löscht Benutzer

verknüpfte Scripte: user.cgi, index.cgi, help.cgi*

[1] Die mit einem Sternchen (*) gekennzeichneten Scripte werden von WebMin bereitgestellt.

1.6 Script: group.cgi

Funktion: Zeigt alle vorhandenen Gruppen, Felder für die Gruppenanlegung, Felder für die Gruppenlöschung, Link für die Gruppenmodifizierung

verknüpfte Scripte: groupadd.cgi, groupmod.cgi, groupdel.cgi, index.cgi, help.cgi*

1.7 Script: groupadd.cgi

Funktion: Legt eine neue Gruppe an

verknüpfte Scripte: group.cgi, index.cgi, help.cgi*

1.8 Script: groupmod.cgi

Funktion: Zeigt Gruppen-Details und modifiziert sekundäre Gruppenmitglieder

verknüpfte Scripte: groupmod.cgi, group.cgi, index.cgi, help.cgi*

1.9 Script: groupdel.cgi

Funktion: Löscht Gruppen

verknüpfte Scripte: group.cgi, index.cgi, help.cgi*

1.10 Script: userauto.cgi

Funktion: Eingeben der Daten für die automatische Benutzergenerierung

verknüpfte Scripte: userauto2.cgi, index.cgi, help.cgi*

1.11 Script: userauto2.cgi

Funktion: Legt neuen Benutzer automatisch an

verknüpfte Scripte: index.cgi, help.cgi*

2 Bibliotheksfunktionen in usermgr-lib.pl

2.1 Funktion: passwdgen()

Übergabewert: String (Name);

Rückgabewert: String (generierter Paßwortstring);

Beschreibung: Diese Funktion generiert einen zufälligen String, der als Paßwort weiterverwendet werden kann. Dabei wird zuerst versucht, den übergebenen Namen mit Hilfe des Soundex-Algorithmus zu packen. Ist dieses erfolgreich, werden die restlichen Zeichen bis zur eingestellten Paßwortlänge (in $maxlength) zufällig erzeugt, ansonsten wird der gesamte Rückgabestring zufällig erzeugt.

2.2 Funktion: getgroups()

Übergabewert: "normal" | "system" (Gruppentyp) [optional];

Rückgabewert: Hash (Gruppen; Gruppe -> GID);

Beschreibung: Liefert die Gruppen mit GID zurück, die im System vorhanden sind. Wird der optionale String "normal" übergeben, werden nur Gruppen mit einer GID größer 99 und kleiner 65534 zurückgeliefert, beim String "system" nur Gruppen mit einer GID kleiner 100 oder gleich 65534.

2.3 Funktion: getusers()

Übergabewert: "normal" | "system" (Benutzertyp) [optional];

Rückgabewert: Hash (User; User -> UID);

Beschreibung: Liefert die User mit UID zurück, die im System vorhanden sind. Wird der optionale String "normal" übergeben, werden nur User mit einer UID größer 99 und kleiner 65534 zurückgeliefert, beim String "system" nur User mit einer UID kleiner 100 oder gleich 65534.

2.4 Funktion: getusergroups()

Übergabewert: String (User);

Rückgabewert: Hash (Gruppen; Gruppe -> GID);

Beschreibung: Liefert die Gruppen mit GID zurück, in der der übergebene User sekundäres Mitglied ist.

2.5 Funktion: getgroupsecuser()

Übergabewert: String (Gruppe);

Rückgabewert: Hash (User; User -> UID);

Beschreibung: Liefert die User mit UID zurück, die in der übergebenen Gruppe sekundäres Mitglied sind.

2.6 Funktion: getgroupprimuser()

Übergabewert: String (Gruppe);

Rückgabewert: Hash (User; User -> UID);

Beschreibung: Liefert die User mit UID zurück, die in der übergebenen Gruppe primäres Mitglied sind.

2.7 Funktion: readpopmultiple()

Übergabewert: String (Wert aus Listenfeld);

Rückgabewert: Array (Listeneinträge);

Beschreibung: Liefert die einzelnen Listeneinträge als Array zurück, die im übergebenen String mit "\0" getrennt waren.

3 Weitere Dateien und Verzeichnisse

3.1 Datei config

Funktion: Beinhaltet die Werte für die Konfigurationsparameter, wie sie bei einer Neuinstallation gesetzt werden.

3.2 Datei config.info.xx

Funktion: Beinhaltet die Parameter für die Modul-Konfiguration in der jeweiligen Sprache.

3.3 Datei module.info

Funktion: Beinhaltet Modulinformationen wie unterstützte Distributionen, Modul-Namen . . .

3.4 Verzeichnis help

Funktion: In diesem Verzeichnis befinden sich die Hilfeseiten.

3.5 Verzeichnis images

Funktion: Die einzelnen Graphiken, die für das Modul verwendet werden, werden in diesem Verzeichnis abgelegt.

3.6 Verzeichnis lang

Funktion: Für die Internationalisierung werden alle Texte und String in Dateien unter diesem Verzeichnis abgelegt, dabei gibt es für jede Sprache eine eigene Datei.

Anlage IV

Quellcode

```
#!/usr/bin/perl
########################################################################
######
#Script:         index.cgi
#Funktion:       Startseite für Benutzer/Gruppenmanager generieren
#                Prüfung der Konfigurationseinstellungen
#Aufgerufen von: WebMin−Menü
#                user.cgi
#                group.cgi
#                userauto.cgi
#Konfigurations−
#variablen:      homedir_schueler (Pfad zu Schülerheimatverzeichnissen
)
#                homedir_lehrer   (Pfad zu Lehrerheimatverzeichnissen)
#                groupdir         (Pfad zu Gruppenverzeichnissen)
#                skelshell        (Pfad zu Profildateien [mit Shell])
#                skelnoshell      (Pfad zu Profildateien [ohne Shell])
#                homedir_perms    (Zugriffsrechte eines neuen Heimatve
rz.)
#                groupdir_perms   (Zugriffsrechte eines neuen Gruppenv
erz.)
#                userrow          (Spaltenanzahl der Benutzerlisten)
#                grouprow         (Spaltenanzahl der Gruppenlisten)

#                stdshell         (Pfad/Dateiname zur Standardshell)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 20.02.2001
########################################################################
######

########################################################################
######
#Verwendete Variablen (local)
#
my ($fehler,              #Zählvariable für Konfigurationsfehler
    $progfehler           #Zählvariable für Programmfindefehler
   )=0;
my @programms=(’chown’,     #Liste der verwendeten Systemprogramme
               ’mkdir’,
               ’groupadd’,
               ’groupdel’,
               ’groupmod’,
               ’useradd’,
               ’userdel’,
               ’usermod’,
               ’smbpasswd’,
               ’recode’,
               ’cp’,
              );
my @links=("user.cgi",                #Linkliste für Icon−Leiste
           "group.cgi",
           "userauto.cgi",
          );
my @icons=("./images/user.gif",       #Iconnamen für Icon−Leiste
           "./images/group.gif",
           "./images/userauto.gif"
          );
my @titles=();                        #Titel für Icon−Leiste
```



```
#
########################################################################
######
########################################################################
######
#Bibliotheken einbinden
#
do ’../web−lib.pl’;
&init_config();

########################################################################
######
#Start der HTML−Ausgabe
#
&header($text{titel},"","usermgr",1,1,undef, "<nobr>Written by</nobr><br>
    <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
    <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
    <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p><b>\n";
########################################################################
######
#Prüfung der Konfigurationseinstellungen
if ($config{homedir_schueler} eq ""
           or not −d $config{homedir_schueler}){
    print text(index1, $text{index2}) . "<br>\n";
    $fehler++;
}
if ($config{homedir_lehrer} eq ""
           or not −d $config{homedir_lehrer}){
    print text(index1, $text{index3}) . "<br>\n";
    $fehler++;
}
if ($config{groupdir} eq ""
           or not −d $config{groupdir}){
    print text(index1, $text{index4}) . "<br>\n";
    $fehler++;
}
if ($config{skelshell} eq ""
           or not −d $config{skelshell}){
    print text(index1, $text{index5}) . "<br>\n";
    $fehler++;
}
if ($config{skelnoshell} eq ""
           or not −d $config{skelnoshell}){
    print text(index1, $text{index6}) . "<br>\n";
    $fehler++;
}
if ($config{homedir_perms} eq ""
           or $config{homedir_perms} !~ /^7[0|5|7]{2}/){
    print "$text{index7}<br>\n";
    $fehler++;
}
if ($config{groupdir_perms} eq ""
           or $config{groupdir_perms} !~ /^77[0|5|7]$/){
    print "$text{index8}<br>\n";
    $fehler++;
}
```

```
if ($config{userrow} !~ /^\d+$/ or $config{userrow} < 1
           or $config{userrow} > 10){
    print "$text{index9}<br>\n";
    $fehler++;
}
if ($config{grouprow} !~ /^\d+$/ or $config{grouprow} < 1
           or $config{grouprow} > 10){
    print "$text{index10}<br>\n";
    $fehler++;
}
if ($config{stdshell} eq "" or not -x $config{stdshell}){
    print "$text{index11}<br>\n";
    $fehler++;
}
########################################################################
######
#Überprüfung, ob alle benötigten Systemprogramme ausgeführt werden könn
en
foreach (@programms){
    unless (has_command($_)){
        print text(index12, $_) . "<br>\n";
        $progfehler++;
    }
}
########################################################################
######
#Ausgabe des Menüs, falls Konfiguration korrekt und alle Systemprogramm
e
#vorhanden sind,
#andernfalls die Anzahl der Fehler und Hinweis auf die Konfiguration
if ($fehler == 0 and $progfehler == 0){
    @titles=($text{image1},$text{image2},$text{image3});
    icons_table(\@links, \@titles, \@icons, 3);
}
else {
    if ($fehler){
        print "<hr>\n";
        print "<h3>" . text(index13, $fehler,
                   "<a href=\"/config.cgi?$module_name\">", "</a>") . "</h3>";
    }
    elsif ($progfehler){
        print "<hr>\n";
        print "<h3>" . text(index14, $progfehler) . "</h3>";
        print text(index15, $gconfig{path});
    }
}
print "</b><hr>\n";
&footer("/", "Webmin index");
```



```
#!/usr/bin/perl
########################################################################
######
#Script:           user.cgi
#Funktion:         Zeigt alle vorhandenen Benutzer
#                  Felder für die Benutzeranlegung
#                  Felder für die Benutzerlöschung
#                  Link für die Benutzermodifizierung
#Konfigurations-
#variablen:        userrow        (Anzahl der Tabellen-Spalten)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
########################################################################
######

########################################################################
######
#Verwendete Variablen (local)
#
my $temp1 = 0;   #Laufvariable für Arrayzugriff
my @temp = ();   #Temporäres Array für Gruppen
my (%usernor,    #alle normalen Benutzer (User -> UID)
    %usersys     #alle System-Benutzer (User -> UID)
   )=();
#
########################################################################
######
########################################################################
######
#Bibliotheken einbinden
#
do './usermgr-lib.pl';
do '../web-lib.pl';
&init_config();

%usersys = getusers('system');
%usernor = getusers('normal');

########################################################################
######
#Start der HTML-Ausgabe
#
&header($text{titelu1},"","user",undef, undef,undef, "<nobr>Written by</nobr><
br>
      <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
      <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
      <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p>\n";
########################################################################
######
#Beginn Formular
#Aufrufscript: useradd.cgi
#Method:       post
#Felder:       txt_username (Accountname)
#              txt_realname (Realer Name)
#              pop_group    (primäre Gruppe)
#              rad_passwd   (Paßwort generieren [yes|no])
```

```
#               rad_usertyp  (Benutzertyp [s|t])
#               rad_shell    (Shell zuweisen [/bin/false|stdshell])
#               rad_smb      (Samba-Account [yes|no])
#               rad_skel     (Heimatverzeichnis erstellen [yes|no])
print "<form action=./useradd.cgi method=post>\n";
########################################################################
######
#Ausgabe der Tabelle für das Anlegen einers neuen Benutzers
print "<h3>$text{user3}</h3>\n";
print "<table border=0 width=100% $cb cellpadding=5 cellspacing=0>\n";
print "<tr>\n";
print "<td><b>" . hlink($text{user4}, "user#1") . ":</b></td>\n";
print "<td><input type=text maxlength=15 name=txt_username size=20></td>\n";
print "<td><b>" . hlink($text{user5}, "user#2") . ":</b></td>\n";
print "<td><input type=text maxlength=50 name=txt_realname size=25></td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td><b>" . hlink($text{user17}, "user#3") . ":</b></td>\n";
print "<td><select name=pop_group size=1>\n";
print "<option selected>------</option>\n";
%groups = &getgroups("normal");
foreach (sort keys %groups){
    print "<option>$_</option>\n";
}
print "</select>\n";
print "</td>\n";
print "<td><b>" . hlink($text{user16}, "user#4") . ":</b></td>\n";
print "<td><input type=radio name=rad_passwd value=yes checked> $text{user13}<br>\n";
print "<input type=radio name=rad_passwd value=no> $text{user14}\n";
print "</td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td><b>" . hlink($text{user6}, "user#5") . ":</b></td>\n";
print "<td><input type=radio name=rad_usertyp value=s checked> $text{user7}<br>\n";
print "<input type=radio name=rad_usertyp value=t> $text{user8}\n";
print "</td>\n";
print "<td><b>" . hlink($text{user9}, "user#6") . ":</b></td>\n";
print "<td><input type=radio name=rad_shell value=\"\/bin\/false\" checked> $text{user10}<b
r>\n";
print "<input type=radio name=rad_shell value=$config{stdshell}> $text{user11} $config{stds
hell})\n";
print "</td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td><b>" . hlink($text{user12}, "user#7") . ":</b></td>\n";
print "<td><input type=radio name=rad_smb value=yes checked> $text{user13}<br>\n";
print "<input type=radio name=rad_smb value=no> $text{user14}</br>\n";
print "</td>\n";
print "<td><b>" . hlink($text{user15}, "user#8") . ":</b></td>\n";
print <<EOP;
<td><input type=radio name=rad_skel value=yes checked> $text{user1
3}<br>
<input type=radio name=rad_skel value=no> $text{user14}
</td>
</tr>
</table>
<br>
<table border=0 cellspacing=0 cellpadding=5>
```



```
<tr>
<td> <input type=submit value="$text{submitu1}"> </td>
<td> <input type=reset value="$text{reset}"> </td>
</tr>
</table>
</form>
<hr>
EOP
#Formular- und Tabellenende
########################################################################
######
########################################################################
######
#Beginn Formular
#Aufrufscript: userdel.cgi
#Method:       post
#Felder:       chk_*       (Benutzername)
#              chks_*      (Systembenutzername)
#              rad_homedel (Heimatverzeichnisse löschen)
#Aufrufscript: usermod.cgi (per Link)
#Methode:      get
#Felder:       user (Benutzername)
#
print "<form action=./userdel.cgi method=post>\n";
print "<h3>$text{user1}</h3>\n";
########################################################################
######
#Ausgabe der Benutzer-Tabelle
print "<table border width=100%>\n";
print "<tr $tb align=center>\n";
for (1..$config{userrow}){
    print "<td><b>$text{user19}</b></td>\n";
    print "<td><b>$text{user20}</b></td>\n";
}
print "</tr>\n";
@temp = sort keys %usernor;
while ($temp1 <= $#temp){
    print "<tr $cb>\n";
    for (1..$config{userrow}){
       print "<td> <a href=./usermod.cgi?user=$temp[$temp1]>$temp[$temp1]</a> 
</td>\n";
       if ($temp[$temp1] ne ""){
           print "<td align=center><input type=checkbox name=chk_$temp[$temp1] value=$te
mp[$temp1]> </td>\n";
       }
       else {
           print "<td> </td>\n";
       }
       $temp1++;
    }
    print "</tr>\n";
}
print <<EOP;
</table>
<br>
<table border=0 $cb cellpadding=5 cellspacing=0>
<tr>
EOP
```

```
print "<td><b>" . hlink($text{user18}, "user#9") . ":</b></td>\n";
print <<EOP;
<td><input type=radio name=rad_homedel checked value=yes> $text{us
er13}<br>
<input type=radio name=rad_homedel value=no> $text{user14}
</td>
</tr>
</table>
<hr>
EOP
#Tabellenende
##########################################################################
######
##########################################################################
######
#Ausgabe der Systembenutzer-Tabelle
#
print "<h3>$text{user2}</h3>\n";
print "<table border width=100%>\n";
print "<tr $tb align=center>\n";
for (1..$config{userrow}){
    print "<td><b>$text{user19}</b></td>\n";
    print "<td><b>$text{user20}</b></td>\n";
}
print "</tr>\n";
$temp1=0;
@temp=();
@temp= sort keys %usersys;
while ($temp1 < $#temp){
    print "<tr $cb>\n";
    for (1..$config{userrow}){
        print "<td> <a href=./usermod.cgi?user=$temp[$temp1]>$temp[$temp1]</a>&nbs
p;</td>\n";
       if ($temp[$temp1] ne ""){
           print "<td align=center><input type=checkbox name=chks_$temp[$temp1] value=$t
emp[$temp1]></td>\n";
       }
       else {
           print "<td> </td>\n";
       }
       $temp1++;
    }
    print "</tr>\n";
}
@temp=();
print "</table>\n";
#Tabellenende
##########################################################################
######
##########################################################################
######
#Tabelle für Buttons
print <<EOT;
<br>
<table border=0 cellspacing=0 cellpadding=5>
  <tr>
    <td> <input type=submit value="$text{submitu2}"> </td>
    <td> <input type=reset value="$text{reset}"> </td>
```



```
  </tr>
</table>
</form>
<hr>
EOT
#
#Formular- und Tabellenende
##########################################################################
######

&footer("/$module_name", $text{back1});
```

```
#!/usr/bin/perl
##########################################################################
######
#Script:          useradd.cgi
#Funktion:        Legt neuen Benutzer an
#Konfigurations−
#variablen:       homedir_schueler (Pfad zu Schülerheimatverzeichnissen
)
#                 homedir_lehrer   (Pfad zu Lehrerheimatverzeichnissen)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
##########################################################################
######

##########################################################################
######
#Verwendete Variablen (local)
#
my ($return)=0;   #Rückgabewert von Systemaufrufen
my ($passwd,      #Paßwort
    $passwdvalid  #Paßwortvalidierung
   )="";
#
##########################################################################
######
##########################################################################
######
#Bibliotheken einbinden
#
do '../web−lib.pl';
do './usermgr−lib.pl';

&init_config();
&ReadParse();

##########################################################################
######
#Funktion:        passwdpart()
#Übergabewert:    keine
#Rückgabewert:    keine
#Beschreibung:    Liefert HTML−Code für die Eingabe eines Paßwortes
#Aufrufscript:    useradd.cgi
#Method:          post
#Felder:          txt_password
#                 txt_passwordvalid
#                 pop_group
#                 rad_passwd
#                 rad_shell
#                 skel
#                 userhome
#                 txt_username
#                 txt_realname
#                 rad_smb
##########################################################################
######
sub passwdpart {
    print <<EOP;
    <form action=./useradd.cgi method=post>
```



```
    <table border=0 cellpadding=5>
    <tr>
    <td><b>
EOP
    print hlink($text{useradd8}, "useradd#$text{useradd8}");
    print <<EOP;
    :</b></td>
    <td><input type=password maxlength=12 name=txt_passwd size=14></td>
    </tr>
    <tr>
    <td><b>
EOP
    print hlink($text{useradd9}, "useradd#$text{useradd9}");
    print <<EOP;
    :</b></td>
    <td><input type=password maxlength=12 name=txt_passwdvalid size=14>
</td>
    </tr>
    <tr>
    <td><input type=submit value=$text{submitua1}></td>
    <td><input type=reset value=$text{reset}></td>
    </tr>
    <tr>
    <td><input type=hidden name=pop_group value=$in{pop_group}>
        <input type=hidden name=rad_passwd value=$in{rad_passwd}>
        <input type=hidden name=rad_shell value=$in{rad_shell}>
        <input type=hidden name=skel value=$in{skel}>
        <input type=hidden name=userhome value=$in{userhome}>
        <input type=hidden name=txt_username value=$in{txt_username}>
        <input type=hidden name=txt_realname value="$in{txt_realname}">
        <input type=hidden name=rad_smb value=$in{rad_smb}>
    </td>
    <td>
    </td>
    </tr>
    </table>
    </form>
EOP
}

##########################################################################
######
#Überprüfung der übergebenen Werte und Bildung des String für das Heima
tverz.
if ($in{txt_username} !~ /^\w+$/){
   error($text{fehlerua1});
   exit;
}
if ($in{pop_group} eq "−−−−−−"){
    error($text{fehlerua2});
    exit;
}
if ($in{rad_usertyp} eq "s"){
    $in{userhome} = $config{homedir_schueler} . "/"
       . $in{pop_group} . "/" . $in{txt_username};
}
elsif ($in{rad_usertyp} eq "t"){
    $in{userhome} = $config{homedir_lehrer} . "/" . $in{txt_username}
```

```
}
if ($in{rad_skel} eq "yes" and -e $in{userhome}){
    error(text(fehlerua3, $in{userhome}));
}
if ($in{rad_skel} eq "yes"){
    if ($in{rad_shell} eq $config{stdshell}){
        $in{skel}=$config{skelshell};
    }
    else {
        $in{skel}=$config{skelnoshell};
    }
}
########################################################################
######
#Start der HTML-Ausgabe
#
&header($text{titelu2},"","useradd",undef, undef,undef, "<nobr>Written by</nob
r><br>
    <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
    <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
    <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p>\n";
print "<pre>\n";
########################################################################
######
#Generierung des Paßwortes, Paßwortvalidierung, Ausgabe von HTML-Code
#für das Eingeben des Paßwortes
if ($in{rad_passwd} eq "yes"){
    print "$text{useradd1}:";
    ####################################################################
######
    #Initialisierung des Zufallszahlengenerator und
    #
    srand (time() ^ ($$ + ($$ << 15)));
    $passwd=&passwdgen($in{txt_username});
    print "<b>$text{return1}</b>\n";
    print "$text{useradd2}:<b>$passwd</b>\n";
}
elsif (defined $in{txt_passwd}){
    if ($in{txt_passwd} eq $in{txt_passwdvalid}){
        $passwdvalid="ok";
        $passwd=$in{txt_passwd};
    }
    else {
        print "<h3>$text{useradd10}</h3></pre>\n";
        &passwdpart();
    }
}
else {
    print "</pre>\n";
    &passwdpart();
}
########################################################################
######
#Anlegen des Benutzer und des Samba-Accounts
if ($in{rad_passwd} eq "yes" or $passwdvalid eq "ok"){
    print text(useradd4, $in{txt_username}) . ":";
```



```
    if ($in{rad_skel} eq "yes"){
        $return = system("useradd -g $in{pop_group} -d $in{userhome} -s $in{rad_shell}
-c \"$in{txt_realname}\" -m -k $in{skel} -p $passwd $in{txt_username} >/dev/null 2>&1") / 256;
    }
    else {
        $return = system("useradd -g $in{pop_group} -d $in{userhome} -s $in{rad_shell}
-c \"$in{txt_realname}\" -p $passwd $in{txt_username} >/dev/null 2>&1") / 256;
    }
    if ($return == 0){
        print "<b>$text{return1}</b>\n";
        print "$text{useradd5}: <b>$in{pop_group}</b>\n";
        print "$text{useradd6}: <b>$in{userhome}</b>\n";
    }
    else {
        print "<b>$text{return2} [" .
            hlink("RC = $return","systemprogs#ua$return") . "]</b>\n";
    }
    if ($in{rad_smb} eq "yes" and $return == 0){
        print "$text{useradd7}: ";
        $return = system("smbpasswd -a -e -s $in{txt_username} $passwd >/dev/null 2>&1
") / 256;
        if ($return == 0){
            print "<b>$text{return1}</b>\n";
        }
        else {
            print "<b>$text{return2} [" .
                hlink("RC = $return", "systemprogs#sm$return") . "]</b>\n";
        }
    }
    elsif ($in{rad_smb} eq "yes"){
        print "$text{useradd7}: <b>$text{return3}</b>\n";
    }
    print "</pre>";
}
print "<hr>";
&footer("/$module_name/user.cgi", $text{back3});
```

```
#!/usr/bin/perl
##########################################################################
######
#Script:          usermod.cgi
#Funktion:        Zeigt Benutzer-Details und
#                 modifiziert Benutzerdaten
#Aufgerufen von:  user.cgi
#                   Werte: user              (Benutzername)
#                 usermod.cgi
#                          rad_passchange    (Paßwort ändern)
#                          txt_primgroup     (neue Primäre Gruppe)
#                          pop_secgroupsdel  (aus sek. Gruppen löschen)
#                          pop_secgroupsadd  (zu sek. Gruppen hinzufüge
n)
#                          rad_samba         (Anderung des Samba-Accoun
ts)
#                          pop_shell         (Shell des Benutzers)
#                          txt_passwd        (Paßwort)
#                          txt_passwdvalid   (Paßwortvalidierung)
#                          user              (Benutzername)
#                          shell             (Shell)
#                          primgroup         (Primäre Gruppe)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
##########################################################################
######

##########################################################################
######
#Verwendete Variablen (local)
#
my ($str_usermod,     #Optionsstring für usermod
    $str_smbpasswd,   #Optionsstring für smbpasswd
    $schluessel,      #Laufvariable für foreach-Schleifen
    $primgroup        #Primäre Gruppe des Users
   )="";
my ($change,          #Kontrollvariable, ob Änderungen vorgenommen wurd
en
    $temp1,           #kennzeichnet Änderungen an sek. Gruppen
    $return,          #Variable für Rückgabewerte von Systemaufrufen
   )=0;
my (@user,            #Informationen über den Benutzer
    @delgroups,       #Gruppen, aus denen der Benutzer entfernt werden
soll
    @addgroups,       #Gruppen, zu denen der Benutzer hinzugefügt werde
n soll
   )=();
my (%secgroups,       #alle sekundären Gruppen des Benutzers; Group ->
GID
    %groups           #alle vorhandenen Gruppen; Group -> GID
   )=();
#
##########################################################################
######
##########################################################################
######
#Bibliotheken einbinden
#
```



```
do '../web-lib.pl';
do './usermgr-lib.pl';
&init_config();
ReadParse();
##########################################################################
######
#Überprüfung, ob Änderungen vorgenommen werden müssen und Generierung
#der Optionsstrings für die Systemprogramme
#
if (defined $in{primgroup}){
    if ($in{rad_passchange} eq "ja" or $in{rad_samba} eq "add"
           or $in{rad_samba} eq "passwd"){
        if ($in{txt_passwd} ne $in{txt_passwdvalid}){
            error($text{fehlerum1});
            exit;
        }
        if ($in{rad_passchange} eq "ja"){
            $str_usermod="-p \"$in{txt_passwd}\" ";
        }
        if ($in{rad_samba} eq "add"){
            $str_smbpasswd="-a -e -s $in{user} \"$in{txt_passwd}\"";
        }
        elsif ($in{rad_samba} eq "passwd"){
            $str_smbpasswd="-s $in{user} \"$in{txt_passwd}\"";
        }
        $change=1;
    }
    if ($in{rad_samba} eq "enable"){
        $str_smbpasswd="-e -s $in{user}";
        $change=1;
    }
    elsif ($in{rad_samba} eq "disable"){
        $str_smbpasswd="-d -s $in{user}";
        $change=1;
    }
    elsif ($in{rad_samba} eq "delete"){
        $str_smbpasswd="-x -s $in{user}";
        $change=1;
    }
    if ($in{txt_primgroup} ne $in{primgroup}){
        if (getgrnam($in{txt_primgroup}) eq undef){
            error($text{fehlerum2});
            exit;
        }
        else {
            $change=1;
            $str_usermod.="-g $in{txt_primgroup} ";
        }
    }
    if ($in{shell} ne $in{pop_shell}){
        $str_usermod="-s $in{pop_shell}";
        $change=1;
    }
    ######################################################################
######
    #sekundäre Gruppenzugehörigkeiten ändern
    %secgroups=getusergroups($in{user});
    if (defined $in{pop_secgroupsdel}){
```

```
        @delgroups=readpopmultiple($in{'pop_secgroupsdel'});
        foreach (@delgroups){
            delete $secgroups{$_};
        }
        $temp=1;
    }
    if (defined $in{pop_secgroupsadd}){
        @addgroups=readpopmultiple($in{'pop_secgroupsadd'});
        foreach (@addgroups){
            $secgroups{$_}=getgrnam($_);
        }
        $temp=1;
    }
    if ($temp == 1){
        $str_usermod.="-G \"" . join(",", keys %secgroups) . "\" ";
        $change=1;
    }
    undef %secgroups;
}
########################################################################
######
#Start der HTML-Ausgabe
#
&header($text{titelu3},"","usermod",undef,undef,undef, "<nobr>Written by</nobr
><br>
    <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
    <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
    <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p>\n";
########################################################################
######
#Benutzerdaten ändern
if ($change == 1){
    print "<h3>" . text(usermod22, $in{user}) . "</h3>\n";
    print "<pre>";
    if ($str_usermod){
        print "$text{usermod20}: ";
        $return = system("usermod $str_usermod $in{user} >/dev/null 2>&1") / 256;
        if ($return == 0){
            print "<b>$text{return1}</b>\n";
        }
        else {
            print "<b>$text{return2} [" .
                hlink("RC = $return", "systemprogs#um$return") . "]</b>\n";
        }
    }
    if ($str_smbpasswd){
        print "$text{usermod21}: ";
        $return = system("smbpasswd $str_smbpasswd >/dev/null 2>&1") / 256;
        if ($return == 0){
            print "<b>$text{return1}</b>\n";
        }
        else {
            print "<b>$text{return2} [" .
                hlink("RC = $return", "systemprogs#sm$return") . "]</b>\n";
        }
    }
```



```
    print "</pre>\n";
    print "<hr>\n";
}
########################################################################
######
#Tabelle mit Benutzer-Details ausgeben
@user=getpwnam $in{user};
$primgroup=getgrgid $user[3];
print "<h3>" . text(usermod1, $in{user}) . "</h3>\n";
print "<table border=0 cellspacing=0 cellpadding=5 $cb width=100%>\n";
print "<tr>\n";
print "<td>" . hlink($text{usermod2}, "usermod#1") . ":</td>\n";
print "<td><b>$user[0]</b></td>\n";
print "<td>" . hlink($text{usermod3}, "usermod#2") . ":</td>\n";
print "<td><b>$user[2]</b></td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td>" . hlink($text{usermod4}, "usermod#3") . ":</td>\n";
print "<td><b>$user[6]</b></td>\n";
print "<td>" . hlink($text{usermod5}, "usermod#4") . ":</td>\n";
print "<td><b>$primgroup ($user[3])</b></td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td>" . hlink($text{usermod6}, "usermod#5") . ":</td>\n";
print "<td><b>$user[7]</b></td>\n";
print "<td rowspan=2 valign=top>" .
      hlink($text{usermod7}, "usermod#6") . ":</td>\n";
print "<td rowspan=2 valign=top><b>";
%secgroups=&getusergroups($in{user});
foreach $schluessel (sort keys %secgroups){
    print "$schluessel ($secgroups{$schluessel}), ";
}
print " </b></td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td>" . hlink($text{usermod8}, "usermod#7") . ":</td>\n";
print "<td><b>$user[8]</b></td>\n";
print "</tr>\n";
print "</table>\n";
#Tabellenende
########################################################################
######
########################################################################
######
#Beginn Formular
#Aufrufscript: usermod.cgi
#Method: post
#Felder: rad_passchange    (Paßwort ändern)
#        txt_primgroup     (neue Primäre Gruppe)
#        pop_secgroupsdel  (aus sekundären Gruppen löschen)
#        pop_secgroupsadd  (zu sekundären Gruppen hinzufügen)
#        rad_samba         (Anderung des Samba-Accounts)
#        pop_shell         (Shell des Benutzers)
#        txt_passwd        (Paßwort)
#        txt_passwdvalid   (Paßwortvalidierung)
#        user              (Benutzername)   [hidden]
#        shell             (Shell)          [hidden]
#        primgroup         (Primäre Gruppe) [hidden]
```

```
#
print "<form action=./usermod.cgi method=post>\n";
print "<br>\n";
print "<h3>" . text(usermod9, $in{user}) . "</h3>\n";
print "<table border=0 cellspacing=0 cellpadding=2 $cb width=100%>\n";
print "<tr>\n";
print "<td><b>" . hlink($text{usermod10}, "usermod#8") .
      ":</b></td>\n";
print "<td>
   <input type=radio name=rad_passchange value=ja> $text{auswahl1}<br>\n";
print "<input type=radio name=rad_passchange value=nein checked>
   $text{auswahl2}</td>\n";
print "<td><b>" . hlink($text{usermod17}, "usermod#9") .
      ":</b></td>\n";
print "<td><input type=password maxlength=14 name=txt_passwd size=10></td>\n";
print "<td><b>" . hlink($text{usermod18}, "usermod#10") .
      ":</b></td>\n";
print "<td><input type=password maxlength=14 name=txt_passwdvalid size=10>
   </td>\n";
print "<td><b>" . hlink($text{usermod8}, "usermod#11") .
      ":</b></td>\n";
print "<td><select name=pop_shell size=1>\n";
if ($user[8] eq "$config{stdshell}"){
    print "<option selected>$config{stdshell}</option>\n";
    print "<option>/bin/false</option>\n";
}
elsif ($user[8] eq "/bin/false"){
    print "<option selected>/bin/false</option>\n";
    print "<option>$config{stdshell}</option>\n";
}
else {
    print "<option selected>$user[8]</option>\n";
    print "<option>$config{stdshell}</option>\n";
    print "<option>/bin/false</option>\n";
}
print "</select>\n";
print "</td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td><b>" . hlink($text{usermod5}, "usermod#12") .
      ":</b></td>\n";
print "<td>
   <input type=text name=txt_primgroup maxlength=50 size=10 value=$primgroup> "
      . group_chooser_button("txt_primgroup",0,0) . "</td>\n";
print "<td><b>" . hlink($text{usermod24}, "usermod#13") .
      ":</b></td>\n";
print "<td><select name=pop_secgroupsdel size=7 multiple>\n";
foreach $schluessel (sort keys %secgroups){
    print "<option value=\"$schluessel\">
   $schluessel ($secgroups{$schluessel})</option>\n";
}
print "</select></td>\n";
print "<td><b>" . hlink($text{usermod23}, "usermod#14") .
":</b></td>\n";
print "<td><select name=pop_secgroupsadd size=7 multiple>\n";
%groups=&getgroups();
foreach $schluessel (sort keys %groups){
    unless (defined $secgroups{$schluessel} or $schluessel eq $primgrou
```



```
p){
        print "<option value=\"$schluessel\">
     $schluessel ($groups{$schluessel})</option>\n";
    }
}
print "</select></td>\n";
print "<td><b>" . hlink($text{usermod11}, "usermod#15") .
":</b></td>\n";
print <<EOP;
<td><input type=radio name=rad_samba value=none checked> $text{usermod1
2}<br>
<input type=radio name=rad_samba value=add> $text{usermod13}<br>
<input type=radio name=rad_samba value=enable> $text{usermod14}<br>
<input type=radio name=rad_samba value=disable> $text{usermod15}<br>
<input type=radio name=rad_samba value=passwd> $text{usermod10}<br>
<input type=radio name=rad_samba value=delete> $text{usermod16}</td>
</tr>
</table>
<br>
<table border=0 cellspacing=0 cellpadding=5>
<tr>
<td> <input type=submit value=\"$text{submitu3}\"> </td>
<td> <input type=reset value=\"$text{reset}\">
<input type=hidden name=user value=\"$in{user}\">
<input type=hidden name=shell value=\"$user[8]\">
<input type=hidden name=primgroup value=\"$primgroup\"></td>
</tr>
</table>
</form>
<hr>
EOP
#
#Formular und Tabellenende
################################################################################
&footer("/$module_name/user.cgi", $text{back3});
```

```
#!/usr/bin/perl
##########################################################################
######
#Script:          userdel.cgi
#Funktion:        Löscht Benutzer
#Aufgerufen von:  user.cgi
#                   Werte: chk_*       (Benutzernamen)
#                          chks_*      (Systembenutzernamen)
#                          rad_homedel (Heimatverzeichnisse löschen)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
##########################################################################
######

##########################################################################
######
#Verwendete Variablen (local)
#
my ($schluessel,     #Zwischenspeichervariable für foreach−Schleife
    $return,         #Variable für Rückgabewert von Systemaufrufen
    $fehler          #Zähler für Fehler
   )=0;
my @starttime=times; #Startzeitwerte speichern
my @endtime=();      #Array für Endzeitwerte
#
##########################################################################
######
##########################################################################
######
#Bibliotheken einbinden
#
do '../web−lib.pl';
&init_config();
ReadParse();
##########################################################################
######
#Start der HTML−Ausgabe
#
&header($text{titelu4},"","userdel",undef, undef,undef, "<nobr>Written by</nobr
><br>
     <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
     <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
     <nobr>Version <b>$text{version}</b><nobr>");
print <<EOP;
<hr>
<p>
<h3>$text{userdel1}</h3>
<pre>
EOP
##########################################################################
######
#Ausgabe von Fehleranzahl
#
foreach $schluessel (keys %in){
    next if ($schluessel eq "rad_homedel");
    print text(userdel2, $in{$schluessel});
    ######################################################################
######
```



```
    #User und Heimatverzeichnis löschen
    if ($in{rad_homedel} eq "yes" and $schluessel =~ /^chk_/){
        $return = system("userdel −r $in{$schluessel} > /dev/null 2>&1") / 256;
        if ($return == 0){
            print text(userdel3, $text{return1}) . ";";
            print text(userdel4, $text{return1}) . ";";
        }
        else {
            print text(userdel3, $text{return2}) . "<b> [" . hlink("RC =
$return", "systemprogs#ud$return") . "]</b>; ";
            print text(userdel4, $text{return3}) . ";";;
            $fehler++;
        }
    }
    ######################################################################
######
    #nur User löschen
    else {
        $return = system("userdel $in{$schluessel} > /dev/null 2>&1") / 256;
        if ($return == 0){
            print text(userdel3, $text{return1}) . ";";
            print text(userdel4, $text{return3}) . ";";
        }
        else {
            print text(userdel3, $text{return2}) . "<b> [" . hlink("RC =
$return", "systemprogs#ud$return") . "]</b>; ";
            print text(userdel4, $text{return3}) . ";";
            $fehler++;
        }
    }
    ######################################################################
######
    #Samba−Account löschen
    $return = system("smbpasswd −x −t $in{$schluessel} > /dev/null 2>&1") / 256;
    if ($return == 0){
        print text(userdel5, $text{return1}) . ";\n";
    }
    else {
        print text(userdel5, $text{userdel6}) . ";\n";
    }
}
print "</pre>\n";
print "<hr>\n";
##########################################################################
######
#Ausgabe von Fehleranzahl
#
if ($fehler){
    print "<h3>" . text(userdel7, $fehler) . "</h3>\n";
    print "<hr>\n";
}
##########################################################################
######
#Ausgabe von Laufzeitdaten
#
print "<h3>$text{time1}</h3>\n";
@endtime=times;
print "<pre>";
```

```
print text(time2, $endtime[0]−$starttime[0]) . "\n";
print text(time3, $endtime[1]−$starttime[1]) . "\n";
print text(time4, $endtime[2]−$starttime[2]) . "\n";
print text(time5, $endtime[3]−$starttime[3]) . "\n";
print "</pre>\n";
print "<hr>\n";

&footer("/$module_name/user.cgi", $text{back3});
```



```
#!/usr/bin/perl
##########################################################################
######
#Script:           group.cgi
#Funktion:         Zeigt alle vorhandenen Gruppen
#                  Felder für die Gruppenanlegung
#                  Felder für die Gruppenlöschung
#                  Link für die Gruppenmodifizierung
#Konfigurations−
#variablen:        grouprow         (Anzahl der Tabellen−Spalten)
#                  groupdir         (Pfad zu Gruppenverzeichnis)
#                  homedir_schueler (Pfad zu Schülerheimatverzeichnissen
)
#                  homedir_lehrer   (Pfad zu Lehrerheimatverzeichnissen)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
##########################################################################
######

##########################################################################
######
#Verwendete Variablen (local)
#
my (%groupsys, #alle System−Gruppen;   Group −> GID
    %groupusr  #alle normalen Gruppen; Group −> GID
   ) = ();
my (@groupsys, #alle System−Gruppen
    @groupusr  #alle normalen Gruppen
   ) = ();
my $temp1 = 0; #Zählvariable für Array−Zugriff
#
##########################################################################
######
##########################################################################
######
#Bibliotheken einbinden
#
do ’../web−lib.pl’;
do ’./usermgr−lib.pl’;
&init_config();

##########################################################################
######
#alle Gruppen ermitteln
%groupsys=getgroups("system");
%groupusr=getgroups("normal");
@groupsys=sort keys %groupsys;
@groupusr=sort keys %groupusr;
%groupsys=();
%groupusr=();
##########################################################################
######
#Start der HTML−Ausgabe
#
&header($text{titelg1},"","group",undef, undef,undef, "<nobr>Written by</nobr>
<br>
      <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
      <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
```

```
    <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p>\n";
########################################################################
######
#Beginn Formular
#Aufrufscript: groupadd.cgi
#Method:       post
#Felder:       txt_groupname
#              chk_groupdir
#              pop_dir
#
print "<form action=./groupadd.cgi method=post>\n";
########################################################################
######
#Ausgabe der Tabelle für das Anlegen einer neuen Gruppe
print <<EOP;
<h3>$text{group1}</h3>
<table border=0 width=100% $cb cellpadding=5 cellspacing=0>
<tr>
EOP
print "<td><b>" . hlink($text{group2}, "group#1") . ":</b></td>\n";
print <<EOP;
<td><input type=text maxlength=16 name=txt_groupname size=20></td>
</tr>
<tr>
EOP
print "<td><b>" . hlink($text{group3}, "group#2") . ":</b></td>\n";
print <<EOP;
<td><input type=checkbox checked name=chk_groupdir></td>
</tr>
<tr>
EOP
print "<td><b>" . hlink($text{group4}, "group#3") . ":</b></td>\n";
print <<EOP;
<td><select name=pop_dir size=1>
<option value=1>$config{groupdir}/&lt;$text{group7}&gt;</option>
<option selected value=2>
  $config{homedir_schueler}/&lt;$text{group7}&gt;/&lt;$text{group7}&gt;
</option>
<option value=3>$config{homedir_lehrer}/&lt;$text{group7}&gt;</option>
</select>
</td>
</tr>
</table>
<br>
<table border=0 cellspacing=0 cellpadding=5>
<tr>
<td> <input type=submit value="$text{submitg1}"> </td>
<td> <input type=reset value="$text{reset}"> </td>
</tr>
</table>
</form>
<hr>
EOP
#Formular- und Tabellenende
########################################################################
######
```



```
########################################################################
######
#Beginn Formular
#Aufrufscript: groupdel.cgi
#Method:       post
#Felder:       chk_* (Gruppenname)
#
#Aufrufscript: groupmod.cgi (per Link)
#Methode:      get
#Felder:       group (Gruppenname)
#
print "<form action=./groupdel.cgi method=post>\n";
########################################################################
######
#Ausgabe der Benutzergruppen-Tabelle
print "<h3>$text{group5}</h3>\n";
print "<table border width=100%>\n";
print "<tr $tb align=center>\n";
for (1..$config{grouprow}){
    print "<td><b>$text{group7}</b></td>\n";
    print "<td><b>$text{group8}</b></td>\n";
}
print "</tr>\n";
$temp1=0;
while ($temp1 <= $#groupusr){
    print "<tr $cb>\n";
    for (1..$config{grouprow}){
        print "<td> <a href=./groupmod.cgi?group=$groupusr[$temp1]>$groupusr[$temp1
]</a> </td>\n";
       if ($groupusr[$temp1] ne ""){
           print "<td align=center><input type=checkbox name=chk_$groupusr[$temp1] value=
$groupusr[$temp1]></td>\n";
       }
       else {
           print "<td> </td>\n";
       }
       $temp1++;
    }
    print " </tr>\n";
}
print "</table>\n";
print "<hr>\n";
#Tabellenende
########################################################################
######
########################################################################
######
#Ausgabe der Systemgruppen-Tabelle
#
print "<h3>$text{group6}</h3>\n";
print "<table border width=100%>\n";
print "<tr $tb align=center>\n";
for (1..$config{grouprow}){
    print "<td><b>$text{group7}</b></td>\n";
    print "<td><b>$text{group8}</b></td>\n";
}
print "</tr>\n";
$temp1=0;
```

```
while ($temp1 < $#groupsys){
    print "<tr $cb>\n";
    for (1..$config{grouprow}){
        print "<td> <a href=./groupmod.cgi?group=$groupsys[$temp1]>$groupsys[$temp1
]</a> </td>\n";
       if ($groupsys[$temp1] ne ""){
           print "<td align=center><input type=checkbox name=chk_$groupsys[$temp1] value
=$groupsys[$temp1]></td>\n";
       }
       else {
           print "<td> </td>\n";
       }
       $temp1++;
    }
    print "</tr>\n";
}
print "</table>\n";
#Tabellenende
##########################################################################
######
##########################################################################
######
#Tabelle für Buttons
print <<EOP;
<br>
<table border=0 cellspacing=0 cellpadding=5>
<tr>
<td> <input type=submit value="$text{submitg2}"> </td>
<td> <input type=reset value="$text{reset}"> </td>
</tr>
</table>
</form>
<hr>
EOP
#
#Formular- und Tabellenende
##########################################################################
######

&footer("/$module_name", $text{back1});
```



```
#!/usr/bin/perl
##########################################################################
######
#Script:          groupadd.cgi
#Funktion:        Legt eine neue Gruppe an
#Aufgerufen von:  group.cgi
#                   Werte: txt_groupname (Gruppenname)
#                          chk_groupdir  ("on" wenn Gruppenverzeichnis
#                                         angelegt werden soll)
#                          pop_dir       (Angabe, wo das Gruppenverzeic
hnis
#                                         liegen soll)
#Konfigurations-
#variablen:       groupdir_perms   (Zugriffsrechte für Gruppenverzeichn
is)
#                 groupdir         (Pfad zu Gruppenverzeichnis)
#                 homedir_schueler (Pfad zu Schülerheimatverzeichnissen
)
#                 homedir_lehrer   (Pfad zu Lehrerheimatverzeichnissen)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
##########################################################################
######

##########################################################################
######
#Verwendete Variablen (local)
#
my %groups=();  #alle vorhandenen Gruppen; Group -> GID
my ($temp1,     #Pfad für Gruppenverzeichnis
    $return     #Variable für Rückgabewerte von Systemaufrufen
   )=0;
#
##########################################################################
######
##########################################################################
######
#Bibliotheken einbinden
#
do '../web-lib.pl';
do './usermgr-lib.pl';
&init_config();
&ReadParse();
##########################################################################
######
#Überprüfung, ob ein gültiger Gruppenname übergeben wurde
if ($in{txt_groupname}!~ /^\w+$/){
   error($text{fehlerga1});
   exit;
}
##########################################################################
######
#Erzeugung des String, der das Gruppenverzeichnis beinhaltet und
#Überprüfung, ob das Gruppenverzeichnis angelegt werden kann
if ($in{chk_groupdir} eq "on"){
    if ($in{pop_dir} == 1){
        $temp1 = $config{groupdir} . "/" . $in{txt_groupname};
    }
```

```
    elsif ($in{pop_dir} == 3){
        $temp1 = $config{homedir_lehrer} . "/" . $in{txt_groupname};
    }
    else {
        $temp1 = $config{homedir_schueler} . "/" . $in{txt_groupname} .
                 "/" . $in{txt_groupname};
    }
    if (-e $temp1){
        error($text{fehlerga31} . "<i> $temp3 </i>" . $text{fehlerga32});
        exit;
    }
}
#########################################################################
######
#Überprüfung, ob die übergebene Gruppe schon vorhanden ist
%groups=getgroups();
foreach (keys %groups){
    if ($in{txt_groupname} eq $_){
        error($text{fehlerga2});
        exit;}
}
#########################################################################
######
#Start der HTML-Ausgabe
#
&header($text{titelg2},"","groupadd",undef, undef,undef, "<nobr>Written by</no
br><br>
     <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
     <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
     <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p>\n";
print "<pre>\n";
print text(groupadd1, $in{txt_groupname}) . ":";
#########################################################################
######
#Anlegen der Gruppe
$return = system("groupadd $in{txt_groupname} >/dev/null 2>&1") / 256;
if ($return == 0){
    print "<b>$text{return1}</b>\n";
}
else {
    print "<b>$text{return2} [" .
        hlink("RC = $return", "systemprogs#ga$return") . "]</b>\n";
}
#########################################################################
######
#Anlegen des Gruppenverzeichnisses
if ($in{chk_groupdir} eq "on" and $return == 0){
    print "$text{groupadd2}: ";
    $return = system("mkdir -p -m $config{groupdir_perms} $temp1 >/dev/null 2>&1") /
256;
    if ($return == 0){
        print "<b>$text{return1} ($temp1)</b>\n";
        print "$text{groupadd3}: ";
        $return = system("chown .$in{txt_groupname} $temp1 >/dev/null 2>&1") / 256
;
        if ($return == 0){
```



```
            print "<b>$text{return1}</b>\n";
        }
        else {
            print "<b>$text{return2} [".
                hlink("RC = $return" ,"systemprogs#cw$return") . "]</b>\n";
        }
    }
    else {
        print "<b>$text{groupadd3} [" .
            hlink("RC = $return", "systemprogs#mk$return") . "] ($temp3)</b>\n";
        print "$text{groupadd3}: </b>$text{return3}</b>\n";
    }
}
elsif ($in{chk_groupdir}) {
    print "$text{groupadd2}: <b>$text{return3}</b>\n";
    print "$text{groupadd3}: <b>$text{return3}</b>\n";
}

print "</pre>\n";
print "<hr>\n";
&footer("/$module_name/group.cgi", $text{back2});
```

```
#!/usr/bin/perl
########################################################################
######
#Script:           groupmod.cgi
#Funktion:         Zeigt Gruppen−Details und
#                  modifiziert sekundäre Gruppenmitglieder
#Aufgerufen von:   group.cgi
#                    Werte: group       (Gruppenname)
#                  groupmod.cgi
#                    Werte: pop_useradd (Liste der User, die zur Gruppe
#                                        hinzugefügt werden sollen)
#                           pop_userdel (Liste der User, die aus der Gru
ppe
#                                        entfernt werden sollen)
#                           group       (Gruppenname)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
########################################################################
######

########################################################################
######
#Verwendete Variablen (local)
#
my (%alluser,     #alle vorhandenen User; User −> UID
    %secuser,     #alle sekundäre Mitglieder der Gruppe; User −> UID
    %primuser,    #alle primären Mitglieder der Gruppe; User −> UID
    %groups       #alle sekundären Gruppen eines Users; Group −> GID
   ) = ();
my (@deluser,     #alle Mitglieder, die aus der Gruppe entfernt werden
sollen
    @adduser      #alle Mitglieder, die zur Gruppe hinzugefügt werden s
ollen
   ) = ();
my ($schluessel,  #Zwischenspeichervariable für foreach−Schleifen
    $str_usermod, #String für "usermod −G"
    $temp1        #Zwischenspeichervariable
   ) = "";
my $return = 0;   #Variable für Rückgabewerte von Systemaufrufen
#
########################################################################
######
########################################################################
######
#Bibliotheken einbinden
#
do '../web−lib.pl';
do './usermgr−lib.pl';

&init_config();
ReadParse();
########################################################################
######
#Start der HTML−Ausgabe
#
&header($text{titelg3},"","groupmod",undef,undef,undef, "<nobr>Written by</no
br><br>
   <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
```



```
   <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
   <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p>\n";

########################################################################
######
#Änderung der Gruppenmitglieder
#
if (defined $in{pop_useradd} or defined $in{pop_userdel}){
    print "<h3>" . text(groupmod9, $in{group}) . "</h3>";
    ####################################################################
######
    #Mitglieder aus Gruppe entfernen
    if (defined $in{pop_userdel}){
        print "<pre>\n";
        @deluser=readpopmultiple($in{pop_userdel});
        foreach $temp1 (@deluser){
            print text(groupmod10,$temp1,$in{group}) . ":";
            %groups= getusergroups($temp1);
            foreach (keys %groups){
                next if ($_ eq $in{group});
                $str_usermod.="$_,";
            }
            chop $str_usermod;
            $return = system("usermod −G \"$str_usermod\" $temp1 >/dev/null 2>&1")
/ 256;
            if ($return == 0){
                print "<b>$text{return1}</b>\n";
            }
            else {
                print "<b>$text{return2} [" .
                    hlink("RC = $return","systemprogs#um$return") . "]</b>\n";
            }
            undef $str_usermod;
            undef %groups;
        }
        print "</pre>\n";
    }
    #
    ####################################################################
######
    ####################################################################
######
    #Benutzer zur Gruppe hinzufügen
    if (defined $in{pop_useradd}){
        print "<pre>\n";
        @adduser=readpopmultiple($in{pop_useradd});
        foreach $temp1 (@adduser){
            print text(groupmod11,$temp1,$in{group}) . ":";
            %groups=getusergroups($temp1);
            $str_usermod=join ',', keys %groups, $in{group};
            $return = system("usermod −G \"$str_usermod\" $temp1 >/dev/null 2>&1")
/ 256;
            if ($return == 0){
                print "<b>$text{return1}</b>\n";
            }
            else {
```

```
                print "<b>$text{return2} [" .
                    hlink("RC = $return","systemprogs#um$return") . "]</b>\n";
            }
            undef %groups;
        }
        print "</pre>\n";
    }
    print "<hr>\n";
    #
    #######################################################################
######
}
#
#######################################################################
######
#######################################################################
######
#Tabelle mit Gruppen−Details ausgeben
print "<h3>" . text(groupmod1, $in{group}) . "</h3>\n";
print "<table border=0 cellspacing=0 cellpadding=5 width=100% $cb>\n";
print "<tr>\n";
print "<td>" . hlink($text{groupmod2},"groupmod#1") .
      ":</b></td>\n";
print "<td><b>$in{group}</b></td>\n";
print "<td>" . hlink($text{groupmod3}, "groupmod#2") .
      ":</b></td>\n";
print "<td><b>" . getgrnam($in{group}) . "</b></td>\n";
print "</tr>\n";
print "<tr>\n";
print "<td>" . hlink($text{groupmod4}, "groupmod#3") .
      ":</b></td>\n";
print "<td><b>\n";
#######################################################################
######
#Primäre Gruppenmitglieder ausgeben
%primuser=getgroupprimuser($in{group});
foreach $schluessel (sort keys %primuser){
    print "$schluessel ($primuser{$schluessel}), \n";
}

print "</b> </td>\n";
print "<td>" . hlink($text{groupmod5}, "groupmod#4") .
      ":</b></td>\n";
print "  <td><b>\n";
#######################################################################
######
#Sekundäre Gruppenmitglieder ausgeben
%secuser=getgroupsecuser($in{group});
foreach $schluessel (sort keys %secuser){
    print "$schluessel ($secuser{$schluessel}), \n";
}

print "</b> </td>\n";
print "</tr>\n";
print "</table>\n";
#Tabellenende
#######################################################################
######
```



```
#######################################################################
######
#Beginn Formular
#Aufrufscript: groupmod.cgi
#Method: post
#Felder: pop_useradd (Liste der User, die zur Gruppe
#                     hinzugefügt werden sollen)
#        pop_userdel (Liste der User, die aus der Gruppe
#                     entfernt werden sollen)
#        group       (Gruppenname) [hidden]
#
print "<form action=./groupmod.cgi method=post>\n";
print "<br>\n";
print "<h3>" . text(groupmod6, $in{group}) . "</h3>\n";
#######################################################################
######
#Tabelle für die Änderung der Gruppenmitglieder
print "<table border=0 cellspacing=0 cellpadding=5 width=100% $cb>\n";
print "<tr valign=top>\n";
print "<td><b>" . hlink($text{groupmod7}, "groupmod#5") .
      ":</b></td>\n";
#######################################################################
######
#Liste mit sekundären Gruppenmitgliedern,
#die aus Gruppe entfernt werden können
print "<td><select name=pop_userdel size=10 multiple>\n";
foreach $schluessel (sort keys %secuser){
    print "<option value=\"$schluessel\">$schluessel ($secuser{$schluessel})
     </option>\n";
}
print "</select>\n";

print "</td>\n";
print "<td><b>" . hlink($text{groupmod8}, "groupmod#6") .
      ":</b></td>\n";
#######################################################################
######
#Liste mit Benutzern, die in die Gruppe eingetragen werden können
print "<td><select name=pop_useradd size=10 multiple>\n";
%alluser=getusers();
foreach $schluessel (sort keys %alluser){
    if (defined $secuser{$schluessel} or defined $primuser{$schluessel}
){
        next;
    }
    print "<option value=\"$schluessel\">$schluessel ($alluser{$schluessel})
     </option>\n";
}
print "</select>\n";

print "</td>\n";
print "</tr>\n";
print "</table>\n";
#Tabellenende
#######################################################################
######
print "<br>\n";
#######################################################################
```

```
######
#Tabelle für Buttons und versteckte Felder
print "<table border=0 cellspacing=0 cellpadding=5>\n";
print "<tr>\n";
print "<td> <input type=submit value=\"$text{submitg3}\"> </td>\n";
print "<td> <input type=reset value=\"$text{reset}\">\n";
print "       <input type=hidden name=group value=$in{group}>\n";
print "</tr>\n";
print "</table>\n";
#Tabellenende
##########################################################################
######
print "</form>\n";
#
#Formularende
##########################################################################
######
print "<hr>\n";
&footer("/$module_name/group.cgi", $text{back2});
```



```
#!/usr/bin/perl
##########################################################################
######
#Script:          groupdel.cgi
#Funktion:        Löscht Gruppen
#Aufgerufen von:  group.cgi
#                   Werte: chk_*        (Gruppenname)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
##########################################################################
######

##########################################################################
######
#Verwendete Variablen (local)
#
my ($schluessel,     #Zwischenspeichervariable für foreach−Schleife
    $return,         #Variable für Rückgabewert von Systemaufrufen
    $fehler          #Zähler für Fehler
   )=0;
my @starttime=times; #Startzeitwerte speichern
my @endtime=();      #Array für Endzeitwerte
#
##########################################################################
######
##########################################################################
######
#Bibliotheken einbinden
#
do '../web−lib.pl';
&init_config();
ReadParse();
##########################################################################
######
#Start der HTML−Ausgabe
#
&header($text{titelg4},"","groupdel",undef, undef,undef, "<nobr>Written by</no
br><br>
    <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
    <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
    <nobr>Version <b>$text{version}</b><nobr>");
print "<hr>\n";
print "<p>\n";
print "<h3>$text{groupdel1}</h3>\n";
print "<hr>\n";
print "<h3>$text{groupdel2}</h3>\n";
print "<pre>\n";
##########################################################################
######
#Löschen der Gruppen
#
foreach $schluessel (keys %in){
    print text(groupdel3, $in{$schluessel}) . ":";
    $return = system("groupdel $in{$schluessel} >/dev/null 2>&1") / 256;
    if ($return == 0){
        print "<b>$text{return1}</b>\n";
    }
    else {
```

```
        $fehler++;
        print "<b>$text{return2} [" . hlink("RC = $return","systemprogs#gd$return") .
 "]</b>\n";
    }
}

print "</pre>\n";
print "<hr>\n";
########################################################################
######
#Ausgabe von Fehleranzahl
#
if ($fehler){
    print "<h3>" . text(groupdel4, $fehler) . "</h3>\n";
    print "<hr>\n";
}
########################################################################
######
#Ausgabe von Laufzeitdaten
#
print "<h3>$text{time1}</h3>\n";
print "<pre>";
@endtime=times;
print text(time2, $endtime[0]-$starttime[0]) . "\n";
print text(time3, $endtime[1]-$starttime[1]) . "\n";
print text(time4, $endtime[2]-$starttime[2]) . "\n";
print text(time5, $endtime[3]-$starttime[3]) . "\n";
print "</pre>\n";
#
########################################################################
######
print "<hr>\n";
&footer("/$module_name/group.cgi", $text{back2});
```



```
#!/usr/bin/perl
########################################################################
######
#Script:          userauto.cgi
#Funktion:        Eingeben der Daten für die automatische Benutzergener
ierung
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
########################################################################
######

########################################################################
######
#Bibliotheken einbinden
#
do '../web-lib.pl';
&init_config();
########################################################################
######
#Start der HTML-Ausgabe
#
&header($text{titela1},"","userauto",undef, undef,undef,"<nobr>Written by</nobr
><br>
    <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
    <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
    <nobr>Version <b>$text{version}</b><nobr>");
########################################################################
######
#Beginn Formular
#Aufrufscript: userauto2.cgi
#Method:       post
#Felder:       txt_import        (Datei mit Benutzerdaten)
#              txt_export        (Pfad für Ausgabedaten)
#              rad_usertyp       (Benutzertyp [s|t])
#              pop_fieldpattern  (Datensatztrennzeichen [;|,|#|:|TAB])
#              pop_importtyp     (Import-Dateiformat [Linux|Windows|Mac
])
#              pop_exporttyp     (Export-Dateiformat [Linux|Windows|Mac
])
#              txt_prefix        (Prefix, der vor die Gruppen angehängt
 wird)
print <<EOP;
<hr>
<p>
<h3>$text{userauto1}</h3>
<form action=./userauto2.cgi method=post>
<table border=0 cellpadding=5 cellspacing=0 $cb width=100%>
<tr>
EOP
print "<td><b>" . hlink($text{userauto2}, "userauto#1") .
    ":</b></td>\n";
print "<td><input type=text maxlength=100 name=txt_import size=40> \n";
print &file_chooser_button("txt_import",0);
print "</td></tr><tr>\n";
print "<td><b>" . hlink($text{userauto3}, "userauto#2") .
    ":</b></td>\n";
print "<td><input type=text maxlength=100 name=txt_export size=40> \n";
print &file_chooser_button("txt_export",1);
```

```
print "</td></tr><tr>\n";
print "<td><b>" . hlink($text{userauto4}, "userauto#3") .
    ":</b></td>\n";
print <<EOP;
<td>
<input type=radio name=rad_usertyp value=s checked> $text{userauto
8}<br>
<input type=radio name=rad_usertyp value=t> $text{userauto9}
</td></tr><tr>
EOP
print "<td><b>" . hlink($text{userauto5}, "userauto#4") .
    ":</b></td>\n";
print <<EOP;
<td><select name=pop_fieldpattern size=1>
<option value="," selected>  ,  </option>
<option value=";">  ;  </option>
<option value="#">  #  </option>
<option value=":">  :  </option>
</select>
</td>
</tr>
<tr>
EOP
print "<td><b>" . hlink($text{userauto6}, "userauto#5") .
    ":</b></td>\n";
print <<EOP;
<td><select name=pop_importtyp size=1>
<option selected value=win>Windows</option>
<option value=dos>DOS</option>
<option value=mac>MAC</option>
<option value=linux>Linux</option>
</select>
</td>
</tr>
<tr>
EOP
print "<td><b>" . hlink($text{userauto7}, "userauto#6") .
    ":</b></td>\n";
print <<EOP;
<td><select name=pop_exporttyp size=1>
<option selected value=win>Windows</option>
<option value=dos>DOS</option>
<option value=mac>MAC</option>
<option value=linux>Linux</option>
</select>
</td>
</tr>
<tr>
EOP
print "<td><b>" . hlink($text{userauto10}, "userauto#7") .
    ":</b></td>\n";
print <<EOP;
<td><input type=text name=txt_prefix size=40 maxlength=15></td>
</table>
<br>
<table border=0 cellspacing=0 cellpadding=5>
<tr>
<td> <input type=submit value="$text{submitau1}"> </td>
```



```
<td><input type=reset value=$text{reset}> </td>
</tr>
</table>
</form>
<hr>
EOP
#
#Formular und Tabellenende
#########################################################################
######
&footer("/$module_name", $text{back1});
```

```
#!/usr/bin/perl
#########################################################################
######
#Script:          userauto2.cgi
#Funktion:        Legt neuen Benutzer automatisch an
#Konfigurations-
#variablen:       homedir_schueler (Pfad zu Schülerheimatverzeichnissen
)
#                 homedir_lehrer   (Pfad zu Lehrerheimatverzeichnissen)
#Aufgerufen von:  userauto.cgi
#                 Werte: txt_import        (Datei mit Benutzerdaten)
#                        txt_export        (Pfad für Ausgabedaten)
#                        rad_usertyp       (Benutzertyp [s|t])
#                        pop_fieldpattern  (Datensatztrennzeichen
#                                           [;|,|#|:|TAB])
#                        pop_importtyp     (Import−Dateiformat
#                                           [Linux|Windows|Mac])
#                        pop_exporttyp     (Export−Dateiformat
#                                           [Linux|Windows|Mac])
#                        txt_prefix        (Prefix, der vor die Gruppen

#                                           angehängt wird)
#Geschrieben von: Tobias Brinkert
#letzte Änderung: 28.02.2001
#########################################################################
######

#########################################################################
######
#Verwendete Variablen (local)
#
my ($tempname1,           #Temporärer Dateiname 1
    $tempname2,           #Temporärer Dateiname 2
    $stdline,             #Aktuelle Zeile der Eingabedate
    $schluessel,          #Zwischenspeichervariable für foreach−Schleif
en
    $username,            #Generierter Accountname des Users
    $passwd,              #Generiertes Paßwort des Users
    $userhome,            #Heimatverzeichnis des Users
    $exportname           #Pfad/Dateiname für die Datei mit den Benutze
rdaten
   )="";
my ($return,              #Return−Wert von System−Aufrufen
    $fehler,              #Kennzeichnung von Fehlern
    $nr1,                 #Laufvariable 1
    $nr2,                 #Laufvariable 2
    $userall,             #Zähler, wieviele Benutzer angelegt wurden
    $userallsmb           #Zähler, wieviele Samba−Benutzer angelegt wur
den
   )=0;
my $vname=0;              #Stelle, wo sich im Datensatz der Vorname bef
indet
my $nname=1;              #Stelle, wo sich im Datensatz der Nachname be
findet
my $ugroup=2;             #Stelle, wo sich im Datensatz die Gruppe befi
ndet
my @starttime=times;      #Startzeit des Scriptes sichern
my (@backup,              #Sicherung des Datensatzes
```



```
    @temp,                #Temporäres Array
    @endtime              #Endzeit des Scriptes
   )=();
my %groupmembers=();      #Benötigte Gruppen; Gruppe −> Anzahl
#
#########################################################################
######
#########################################################################
######
#Bibliotheken einbinden
#
do './usermgr−lib.pl';
do '../web−lib.pl';
&init_config();
ReadParse();

#########################################################################
######
#Initialisierung des Zufallszahlengenerator und
#Erzeugung von temporären Dateinamen
#
srand (time() ^ ($$ + ($$ << 15)));
$tempname1=tempname();
$tempname2=tempname();

#########################################################################
######
#Prüfen der Werte für Import− und Export−Dateien
#
if (not −e $in{txt_import} or −d $in{txt_import}){
    error($text{fehlerau1});
    exit;
}
until (−d $in{txt_export}){
    error($text{fehlerau2});
    exit;
}
#########################################################################
######
#Kopieren der Import−Datei ins Temp−Verzeichnis
#
$return= system("cp $in{txt_import} $tempname1 >/dev/null 2>&1") / 256;
until ($return == 0){
    system("rm $tempname1 >/dev/null 2>&1");
    error($text{fehlerau3} . "[RC = $return]");
    exit;
}
#########################################################################
######
#Konvertieren der Import−Datei ins Linux−Format
#
if ($in{pop_importtyp} eq "win"){
    $return = system("recode cp1252..latin1 $tempname1 >/dev/null 2>&1") / 256;
}
elsif ($in{pop_importtyp} eq "dos"){
   $return = system("recode ibmpc..latin1 $tempname1 >/dev/null 2>&1") / 256;
}
elsif ($in{pop_importtyp} eq "mac"){
```

```
   $return = system("recode macce..latin1 $tempname1 >/dev/null 2>&1") / 256;
}
else {
    $return=0;
}
until ($return == 0){
    system("rm $tempname1 2>/dev/null");
    error($text{fehlerau4} . "[RC = $return]");
    exit;
}

########################################################################
######
#Start der HTML-Ausgabe
#
&header($text{titela1},"","userauto2",undef, undef,undef,"<nobr>Written by</nob
r><br>
   <nobr><a href=mailto:t.brinkert\@15bit.de>Tobias Brinkert</a><nobr><br>
   <nobr><a href=http://www.15bit.de>Home://page</a><nobr><br>
   <nobr>Version <b>$text{version}</b><nobr>");
print <<EOP;
<hr>
<p>
<h3>$text{userauto21}</h3>
EOP
########################################################################
######
#Einlesen der Import-Datei
open (INPUT, "$tempname1") or die;
open (OUTPUT, ">$tempname2") or die;
while ($stdline=<INPUT>){
    chomp $stdline;
    @backup = split /$in{pop_fieldpattern}/, $stdline;
    ####################################################################
######
    #Überprüfung, ob Import-Datei im richtigen Format vorliegt
    if ($#backup <=> 2){
        print "<pre><b>$text{userauto22}</b></pre>\n";
        $fehler=1;
        last;
    }
    ####################################################################
######
    #Ersetzen von deutschen Umlauten und ß
    #Löschen von einigen Sonderzeichen
    $stdline=~s/ö/oe/g;
    $stdline=~s/ä/ae/g;
    $stdline=~s/ü/ue/g;
    $stdline=~s/Ö/Oe/g;
    $stdline=~s/Ä/Ae/g;
    $stdline=~s/Ü/Ue/g;
    $stdline=~s/ß/ss/g;
    $stdline=~s/\.|\-|\ //g;
    @temp=split /$in{pop_fieldpattern}/, $stdline;
    $temp[2]=~s/([a-z0-9]+)/$1/gi;
    ####################################################################
######
    #wenn Gruppenprefix angegeben, wird der Gruppe das Prefix vorangest
```



```
ellt
    #und die Gruppe in zweistelliger Nummer ausgegeben
    if ($in{txt_prefix}){
        if ($temp[2] =~/^\d\w$/){
            $temp[2] = $in{txt_prefix} . "0" . $temp[2];
            $groupmembers{$temp[2]}++;
        }
        else {
            $temp[2] = $in{txt_prefix} . $temp[2];
            $groupmembers{$temp[2]}++;
        }
    }
    else {
        $groupmembers{$temp[2]}++;
    }
    ####################################################################
######
    #schreibe erzeugte Daten in temporäre Datei
    print OUTPUT join("$in{pop_fieldpattern}", $temp[$vname], $temp[$nname],
"$backup[$vname] $backup[$nname]", $temp[$ugroup]) . "\n";
}
close(INPUT);
close(OUTPUT);
########################################################################
######
#Überprüfung, ob alle Gruppen vorhanden sind
if ($fehler == 0){
   print "<pre>";
   foreach $schluessel (sort keys %groupmembers){
      $return = getgrnam("$schluessel");
      if (not defined $return){
         print text("userauto24", $schluessel, "\n");
         $fehler=1;
      }
   }
   if ($fehler == 0){
       print "<b>$text{userauto23}</b>\n";
   }
   print "</pre>\n";
}
print "<hr>\n";
########################################################################
######
#Benutzer anlegen und Benutzerdaten in Datei schreiben
if ($fehler == 0){
    print "<h3>$text{userauto33}</h3>\n";
    print "<pre>\n";
    open(INPUT, "$tempname2");
    open(OUTPUT, ">$tempname1");
    print OUTPUT "# " . localtime(time) . "\n";
    print OUTPUT "#---Dateisyntax---\n";
    print OUTPUT "#GROUP$in{pop_fieldpattern}NAME$in{pop_fieldpattern}ACCOUNT$in{p
op_fieldpattern}PASSWORD$in{pop_fieldpattern}HOMEDIR$in{pop_fieldpattern}STATE\n";
    while (<INPUT>){
        chomp;
        @temp=split /$in{pop_fieldpattern}/, $_;
        print "$temp[2] ";
        ################################################################
```

```
######
        #Generierung des Accountnamen (max. 12 Zeichen)
        foreach $nr2 (1 .. 8){
            $nr1 = 12 − $nr2;
            $username=lc(substr($temp[0],0,$nr2) . substr($temp[1],0,$n
r1));
            $return=getpwnam(lc("$username"));
            last if (not defined $return);
        }
        ########################################################################
######
        #Generierung des Paßwortes
        $passwd=&passwdgen($username);
        ########################################################################
######
        #Generierung des Heimatverzeichnisses
        if ($in{rad_usertyp} eq "s"){
           $userhome = $config{homedir_schueler} . "/"
           . $temp[3] . "/" . $username;
        }
        elsif ($in{rad_usertyp} eq "t"){
            $userhome = $config{homedir_lehrer} . "/" . $username;
        }
        ########################################################################
######
        #Benutzer anlegen
        if (−d $userhome){
            $return = system("useradd −g $temp[3] −d $userhome −s \"/bin/false\" −c \"$te
mp[2]\" −p $passwd $username > /dev/null 2>&1") / 256;
        }
        elsif (−e $userhome){
            $return = $text{userauto25};
        }
        else {
            $return = system("useradd −g $temp[3] −d \"$userhome\" −s \"/bin/false\" −c \"
$temp[2]\" −m −k $config{skelnoshell} −p $passwd $username > /dev/null 2>&1") / 256;
        }
        ########################################################################
######
        #Samba−Benutzer anlegen
        if ($return == 0){
            $userall++;
            $return = system("smbpasswd −a −e −s $username $passwd > /dev/null 2>&1"
) / 256;
            if ($return == 0){
                $return = $text{userauto27};
                $userallsmb++;
            }
            else {
                $return = $text{userauto26}
            }
        }
        else {
            $return = $text{userauto25};
        }
        print "[Account=<b>$username</b>, $text{userauto34}=<b>$temp[3]</b>]:".
            " <b>$return</b>\n";
        print OUTPUT "$temp[3]$in{pop_fieldpattern}$temp[2]$in{pop_fieldpattern}$username
```



```
$in{pop_fieldpattern}$passwd$in{pop_fieldpattern}$userhome$in{pop_fieldpattern}$return\n";
    }
    print "</pre>\n";
    close(INPUT);
    close(OUTPUT);
    ########################################################################
#######
    #Export−Datei sortieren, ins angegebene Format konvertieren und ins
 ange
    #gebene Verzeichnis kopieren (Name = "userlist_PID.txt")
    system("sort −o $tempname2 $tempname1 >/dev/null 2>&1");
    if ($in{pop_exporttyp} eq "win"){
       system("recode latin1..cp1252 $tempname2 >/dev/null 2>&1");
    }
    elsif ($in{pop_exporttyp} eq "dos"){
       system("recode latin1..ibmpc $tempname2 >/dev/null 2>&1");
    }
    elsif ($in{pop_exporttyp} eq "mac"){
       system("recode latin1..macce $tempname2 >/dev/null 2>&1");
    }
    $exportname= "$in{txt_export}/userlist_$$.txt";

    system("cp −b $tempname2 $exportname");
    print "<hr>\n";
    ########################################################################
#######
    #Gruppenzusammenfassung anzeigen
    print "<h3>$text{userauto28}</h3>\n";
    print "<pre>\n";
    foreach $schluessel (sort keys %groupmembers){
       print text(userauto29, $schluessel, $groupmembers{$schluessel})
. "\n";
    }
    print "</pre>\n";
    print "<hr>\n";
    print "<h3>$text{userauto30}</h3>\n";
    print "<pre>";
    print text(userauto31, $userall) . "\n";
    print text(userauto32, $userallsmb) . "\n";
    print "</pre>\n";
    print "<hr>\n";
    ########################################################################
#######
    #Name der Export−Datei anzeigen
    print "<h3>$text{userauto35}</h3>\n";
    print "<pre>";
    print "$text{userauto36}: <b>$exportname</b>\n";
    print "</pre>";
    print "<hr>\n";
}
########################################################################
#######
#Temporäre Dateien löschen
system("rm $tempname1 2>/dev/null");
system("rm $tempname2 2>/dev/null");
########################################################################
######
#Ausgabe von Laufzeitdaten
```

```
#
print "<h3>$text{time1}</h3>\n";
print "<pre>";
@endtime=times;
print text(time2, $endtime[0]−$starttime[0]) . "\n";
print text(time3, $endtime[1]−$starttime[1]) . "\n";
print text(time4, $endtime[2]−$starttime[2]) . "\n";
print text(time5, $endtime[3]−$starttime[3]) . "\n";
print "</pre>\n";
print "<hr>\n";
&footer("/$module_name", $text{back1});
```



```
titel=Benutzer/Gruppenmanager

titelg1=Gruppenmanager
titelg2=Gruppe anlegen
titelg3=Gruppe modifizieren
titelg4=Gruppe löschen

titelu1=Benutzermanager
titelu2=Benutzer anlegen
titelu3=Benutzer modifizieren
titelu4=Benutzer löschen

titela1=Mehrere Benutzer anlegen
titela2=Mehrere Benutzer anlegen


image1=Benutzermanager
image2=Gruppenmanager
image3=Automatische Benutzeranlegung

imgalt1=Benutzer
imgalt2=Gruppe
imgalt3=Auto−Benutzer

return1=erfolgt
return2=gescheitert
return3=ausgelassen

auswahl1=ja
auswahl2=nein

back1=Übersicht
back2=Gruppenmanager
back3=Benutzermanager
back4=Konfiguration für die automatische Benutzeranlegung

index1=Sie haben keinen oder einen nicht existierenden Pfad zu den $1 a
ngegeben.
index2=Schüler−Heimatverzeichnissen
index3=Lehrer−Heimatverzeichnissen
index4=Gruppenverzeichnissen
index5=Profildateien für neue Benutzer, die eine Shell zugewiesen bekom
men,
index6=Profildateien für neue Benutzer, die keine Shell zugewiesen beko
mmen,
index7=Zugriffsrechte für neue Heimatverzeichnisse fehlerhaft, es sind
nur die Werte 700,750,755,770,775 und 777 erlaubt.
index8=Zugriffsrechte für neue Gruppenverzeichnisse fehlerhaft, es sind
 zur die Werte 770, 775 und 777 erlaubt.
index9=Die Spaltenanzahl für die Benutzertabellen muß eine Zahl größer
0 sein.
index10=Die Spaltenanzahl für die Gruppentabellen muß eine Zahl größer
0 sein.
index11=Sie haben keine oder eine nicht ausführbare Datei als Standards
hell angegeben.
index12=Systemprogramm <i>$1</i> im Pfad nicht gefunden.
index13=Es gibt <i>$1</i> Fehler in der Konfiguration. $2 Bitte konfigu
rieren Sie das Modul neu $3.
```

```
index14=Es gibt <i>$1</i> benötigte Systemprogramme, die im Suchpfad nicht gefunden werden konnten. Bitte überprüfen Sie ihr System.
index15=Suchpfad: $1

user1=Vorhandene Benutzer
user2=Vorhandene Systembenutzer
user3=Neuen Benutzer anlegen
user4=Accountname
user5=Realer Name
user6=Typ des Benutzers
user7=Schüler
user8=Lehrer
user9=Anmeldung direkt auf Server
user10=nicht erlaubt
user11=erlaubt (Standardshell:
user12=Samba-Benutzer einrichten
user13=ja
user14=nein
user15=Heimatverzeichnis erstellen
user16=Paßwort automatisch generieren
user17=Primäre Gruppe
user18=Heimatverzeichnis löschen
user19=Benutzer
user20=Löschen

useradd1=Paßwort wird generiert
useradd2=Das Paßwort lautet
useradd3=Die beiden angegebenen Paßwörter stimmen nicht überein. Bitte wiederholen Sie Ihre Eingaben.
useradd4=Benutzer <i>$1</i> wird angelegt
useradd5=Die primäre Gruppe ist
useradd6=Das Heimatverzeichnis befindet sich unter
useradd7=Samba-Benutzer wird eingerichtet
useradd8=Paßwort eingeben
useradd9=Paßwortwiederholung

userdel1=Angegebene Benutzer werden gelöscht
userdel2=Benutzer <b>$1</b>:
userdel3=aus System löschen=<b>$1</b>
userdel4=Heimatverzeichnis löschen=<b>$1</b>
userdel5=aus Samba löschen=<b>$1</b>
userdel6=nicht vorhanden
userdel7=Es sind beim Löschen <b>$1</b> Fehler aufgetreten.

usermod1=Benutzer-Details für <i>$1</i>
usermod2=Accountname
usermod3=Benutzer ID (uid)
usermod4=Realer Name
usermod5=Primäre Gruppe
usermod6=Heimatverzeichnis
usermod7=sekundäre Gruppen
usermod8=Shell
usermod9=Benutzer-Daten ändern für <i>$1</i>
usermod10=Paßwort ändern
usermod11=Samba<br>Account
usermod12=keine Änderung
usermod13=anlegen und aktivieren
usermod14=aktivieren
```



```
usermod15=deaktivieren
usermod16=löschen
usermod17=Paßwort
usermod18=Paßwort-<br>wiederholung
usermod19=Shell
usermod20=Benutzer-Daten werden geändert
usermod21=Samba-Daten werden geändert
usermod22=Rückmeldung der Benutzermodifizierung von <i>$1</i>
usermod23=zu sekundären<br>Gruppen hinzufügen
usermod24=aus sekundären<br>Gruppen löschen

userauto1=Einstellungen für die automatische Benutzeranlegung
userauto2=Datei mit den Benutzerdaten (mit Pfad)
userauto3=Pfad für die Datei mit den generierten Benutzerdaten
userauto4=Typ der Benutzer
userauto5=Trennzeichen in Benutzerdaten
userauto6=Format der Benutzerdaten
userauto7=Format der generierten Benutzerdaten
userauto8=Schüler
userauto9=Lehrer
userauto10=Prefix für Gruppen
userauto21=Überprüfung, ob alle Gruppen vorhanden sind
userauto22=Die Import-Datei liegt nicht im richtigen Format vor oder sie haben ein verkehrtes Feldtrennzeichen angegeben
userauto23=Erfolgreich abgeschlossen
userauto24=Gruppe <b>$1</b> nicht vorhanden $2
userauto25=nicht angelegt
userauto26=Samba-Benutzer nicht angelegt
userauto27=angelegt
userauto28=Gruppenübersicht
userauto29=Gruppe <b>$1</b>: <b>$2</b> Mitglied(er) hinzugefügt.
userauto30=Übersicht der angelegten Benutzer
userauto31=<b>$1</b> Accounts wurden auf dem Server eingerichtet.
userauto32=<b>$1</b> Accounts wurden für den Samba-Server eingerichtet.
userauto33=Benutzer werden angelegt
userauto34=Gruppe
userauto35=Export-Datei
userauto36=Name/Pfad der Export-Datei

time1=Laufzeit
time2=Das Script benötigte <b>$1</b> CPU-Sekunden.
time3=Das System belegte dabei <b>$1</b> CPU-Sekunden.
time4=Die Child-Prozesse benötigten <b>$1</b> CPU-Sekunden.
time5=Das System belegte dabei <b>$1</b> CPU-Sekunden.

group1=Neue Gruppe anlegen
group2=Gruppenname
group3=Gruppenverzeichnis anlegen
group4=Gruppenverzeichnis
group5=Vorhandene Benutzergruppen
group6=Vorhandene Systemgruppen
group7=Gruppe
group8=Löschen

groupadd1=Gruppe <i>$1</i> wird angelegt
groupadd2=Gruppenverzeichnis wird angelegt
groupadd3=Zugriffsrechte setzen
```

```
groupdel1=Beachten Sie bitte, daß die eventuell vorhandenen Gruppenverz
eichnisse nicht gelöscht werden
groupdel2=Angegebene Gruppen werden gelöscht
groupdel3=Gruppe <i>$1</i> wird gelöscht
groupdel4=Es konnten <b>$1</b> Gruppen nicht gelöscht werden.

groupmod1=Gruppen-Details für <i>$1</i>
groupmod2=Gruppenname
groupmod3=Gruppen-ID (gid)
groupmod4=primäre Gruppenmitglieder
groupmod5=sekundäre Gruppenmitglieder
groupmod6=Sekundäre Gruppenmitglieder für <i>$1</i> ändern
groupmod7=Gruppenmitglied(er) entfernen
groupmod8=Gruppenmitglied(er) hinzufügen
groupmod9=Rückmeldung der Gruppenmodifizierung von <i>$1</i>
groupmod10=Benutzer <b>$1</b> wird aus Gruppe <b>$2</b> entfernt
groupmod11=Benutzer <b>$1</b> wird der Gruppe <b>$2</b> hinzugefügt

tabuser1=Benutzer
tabuser2=Löschen

submitg1=Gruppe anlegen
submitg2=Gruppen löschen
submitg3=Gruppe ändern

submitu1=Benutzer anlegen
submitu2=Benutzer löschen
submitu3=Benutzer ändern
submitua1=Abschicken

submitau1=Benutzer anlegen

reset=Zurücksetzen

fehlerga1=Sie haben keinen oder einen ungültigen Gruppennamen eingegebe
n.
fehlerga2=Die angegebene Gruppe ist schon vorhanden.
fehlerga31=Eine Datei/Verzeichnis mit dem Namen
fehlerga32=existiert bereits. <br>Bitte geben Sie ein anderes Gruppenve
rzeichnis an oder löschen / verschieben Sie die Datei / das Verzeichnis
.

fehlerua1=Sie haben keinen oder einen ungültigen Benutzernamen eingegeb
en
fehlerua2=Sie müssen eine primäre Gruppe auswählen.
fehlerua3=Für den Benutzer wurde als Heimatverzeichnis <i>$1</i> angege
ben. Eine Datei/Verzeichnis mit diesem Namen existiert bereits.<br>Bitt
e geben Sie einen anderen Benutzernamen/Gruppe an oder löschen / versch
ieben Sie die Datei / das Verzeichnis.

fehlerum1=Die beiden angegebenen Paßwörter sind nicht identisch.
fehlerum2=Sie haben eine nicht existierende primäre Gruppe angegeben
fehlerau1=Sie keine oder eine falsche Import-Datei gewählt.
fehlerau2=Sie haben kein oder ein nicht existierendes Export-Verzeichni
s ausgewählt.
fehlerau3=Fehler beim Kopieren der Import-Datei ins TMP-Verzeichnis
fehlerau4=Fehler beim Konvertieren der Import-Datei
```



```
version=0.2.0
```

config.info

Seite 1/1

```
homedir_schueler=Pfad zu Schülerheimatverzeichnissen,8
homedir_lehrer=Pfad zu Lehrerheimatverzeichnissen,8
groupdir=Pfad zu Gruppenverzeichnissen,8
skelshell=Pfad zu Profildateien (mit Shell-Benutzung),8
skelnoshell=Pfad zu Profildateien (ohne Shell-Benutzung),8
homedir_perms=Zugriffsrechte eines neuen Heimatverzeichnisses,1,700-700,750-750,755-755,770-770,775-775,777-777
groupdir_perms=Zugriffsrechte eines neuen Gruppenverzeichnisses,1,770-770,775-775,777-777
userrow=Spaltenanzahl der Benutzerlisten,1,1-1,2-2,3-3,4-4,5-5,6-6,7-7,8-8,9-9,10-10
grouprow=Spaltenanzahl der Gruppenlisten,1,1-1,2-2,3-3,4-4,5-5,6-6,7-7,8-8,9-9,10-10
stdshell=Pfad/Dateiname zur Standardshell,8
```

module.info

Seite 1/1

```
name=UG-Manager
desc=User- and Group-Manager
desc_de=Benutzer- und Gruppen-Manager
os_support=suse-linux
category=system
```

Table of Contents